

Small or medium-scale focused research project (STREP)



**ICT Call 8**  
FP7-ICT-2011-8

## **Cooperative Self-Organizing System for low Carbon Mobility at low Penetration Rates**

The COLOMBO logo, where the letters C, L, M, B, and O are in a bold, black, sans-serif font. The two O's are replaced by circular icons: the first O is red with a black grid pattern, and the second O is green with a black grid pattern. The background of the entire page features a large, faint, circular ripple pattern in shades of green and yellow.

### **COLOMBO: Deliverable D5.4** **Performance of COLOMBO Solutions**

<b>Document Information</b>	
Title	Deliverable 5.4 - Performance of COLOMBO Solutions
Dissemination Level	PU (Public)
Version	1.0
Date	18.11.2015
Status	Approved by project coordinator
Authors	Andreas Leich (Project Coordinator, DLR), Wolfgang Niebel (DLR), Laura Bieker (DLR), Robbin Blokpoel (IMTECH), Federico Caselli (UNIBO), Jérôme Härri (EURECOM), Marek Junghans (DLR), Hagen Saul (DLR), Thomas Stützle (ULB)

## Table of contents

1	Introduction.....	4
1.1	Project Context.....	4
1.2	Document Objectives and Intended Audience .....	5
1.3	Work/Document Structure .....	6
2	Traffic state estimation (TSE) .....	7
2.1	Algorithm 1: Cluster-based .....	8
2.2	Algorithm 2: DissFlow .....	10
2.3	Algorithms 3 and 4: .....	13
2.4	Algorithm 5: Data Fusion.....	13
3	Automatic incident detection (AID) .....	15
4	Emission monitoring System (EMS).....	18
4.1	Basic Approach .....	19
4.2	Clustering Approach.....	19
4.3	Conclusions .....	20
5	Self-organising traffic light control (SOTL).....	21
5.1	Simulation Conditions.....	21
5.1.1	Simulator.....	21
5.1.2	Road Network .....	21
5.1.3	Detectors .....	23
5.1.4	Traffic Conditions and simulation time .....	23
5.1.5	Signal Control .....	25
5.1.6	Simulation System Configuration.....	25
5.2	Simulation results.....	26
5.3	Conclusion and Outlook.....	31
6	Automatic configuration software .....	32
6.1	Simulation setup.....	32
6.1.1	Tuning overview .....	32
6.1.2	Evaluation overview .....	33
6.2	Findings on automatic configuration for the initial design .....	33
6.3	Findings on automatic configuration for online adaptation .....	34
6.3.1	Change of the penetration rate .....	34
6.3.2	Change of TLC parameters .....	34
6.3.3	Change of the traffic density.....	34
6.4	Extended experiments .....	34
6.5	Results .....	35
6.5.1	Same tuning on different networks .....	36
6.5.2	Tuning budget influence .....	39



7    Communication Integration .....42

8    References.....44

Appendix A –Tuning Results.....45

Appendix B – SWARM algorithm flow charts.....53

# 1 Introduction

## 1.1 Project Context

Traffic light control (TLC) as part of active traffic management aims at enabling safe and efficient passing of traffic flows at intersections, thereby reducing fuel consumption and CO<sub>2</sub> emissions. It requires determining the situation on the roads. These both actions make use of technologies which are becoming part of intelligent transportation systems (ITS).

Emerging cooperative techniques (C-ITS) like vehicle-to-infrastructure (V2I) communication via the IEEE 802.11-2012<sup>1</sup> standard may increase the knowledge about the traffic situation and open new channels for delivering information. However, most cooperative applications require large penetration rates in order to assure their functionality<sup>2</sup> making the first steps towards their deployment unattractive.

COLOMBO works on overcoming this hurdle by delivering a set of modern, self-organising traffic management components being applicable even at low OBU penetration rates below 10%, ensuring their usability from the very beginning of deployment on.

COLOMBO focuses on two traffic management topics: traffic surveillance, and advanced traffic light control algorithms. It delivers the methodologies and software for the following urban traffic management components, which are designed to operate locally on road site units (RSUs):

- A cooperative traffic state monitoring system (WP1), including
  - Traffic state estimation (TSE);
  - Automatic incident detection (AID);
  - Emissions monitoring system (EMS);
- A cost-effective, adaptive, and self-organising traffic light (SOTL) control (WP 2), applying swarm intelligence methods, and using data from the WP 1 monitoring system instead of stationary detection;
- A tool for automatic TLC algorithm configuration and tuning (WP 3) to optimize the SOTL of WP2.

To allow the ex-ante appraisal of the solutions' performance and CO<sub>2</sub> emission impacts in this deliverable, COLOMBO extends an available, well-established model and software for simulating vehicular pollutant emissions (PHEM) in WP 4. To take the vehicular population in the year 2020 into consideration, electrical vehicles are included. The implementation into the traffic simulation SUMO as PHEMlight, as well as extracting guidelines on

- Emission-optimal driver behaviour adapted to the new traffic light control system, and
- Developing eco-friendly TLC, dedicated for traffic engineers

are also part of WP4.

The appraisal reported in this deliverable is based solely on results of the simulators for road traffic (SUMO) and vehicular communication (iTETRIS and ns-3), which were extended within WP5, too. They are described in the deliverables "Prototype of overall System Architecture and Definition of Interfaces" [COLOMBO D5.1, 2013] and "Traffic Simulation Extensions" [COLOMBO D5.2, 2014]. Most of the used scenarios as well as the planned appraisal methodology were laid out in "Scenario Specifications and Required Modifications to Simulation Tools" [COLOMBO D1.1, 2014] and "Traffic Light Algorithm Evaluation System" [COLOMBO D5.3, 2014].

<sup>1</sup> The former IEEE 802.11p extension for automotive application was merged into the basic standard in 2012.

<sup>2</sup> Exemplary, a Green Light Optimised Speed Advisory (GLOSA) can yield in waiting time reductions of 1..9% already at 10% penetration rate [eCoMove, 2013] and in reductions of ~15% at 20% OBUs [Katsaros et al., 2011]. Fuel and emission savings start at higher penetration rates between 30% [eCoMove, 2013] and 50% [Katsaros et al., 2011].

The interaction between these work packages is shown in Figure 1.

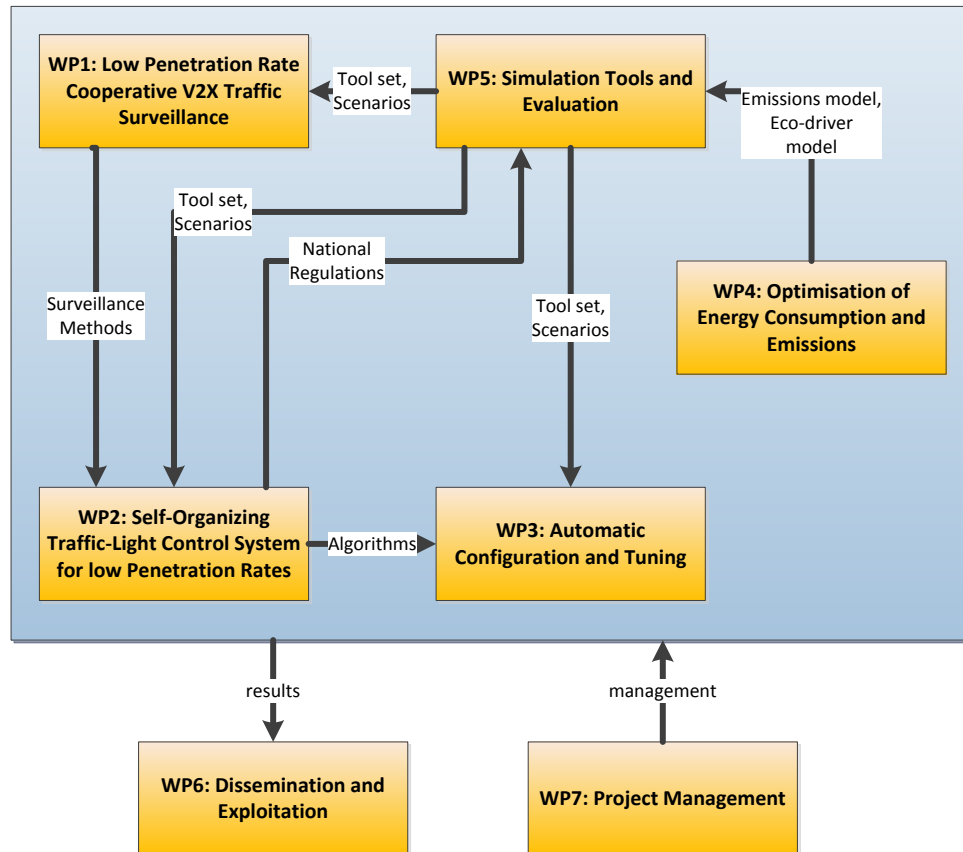


Figure 1: COLOMBO work packages

## 1.2 Document Objectives and Intended Audience

This document shall enable a concise overview on previously reported results of tests and evaluations from different WPs and respective deliverables. It thus avoids searching through a bigger number of different documents. The auxiliary simulation software of WP4 and WP5 is not part of the investigations described herein.

The most comprehensive evaluation of the self-organising traffic light solution (SOTL) was performed on a newly edited real-world scenario (set) comprising a road stretch in the north of the Italian city Monza. This set-up is described in section 5.1 to allow reproducibility and to make it public as open data to interested third parties, mainly other researchers.

The simulation results of the pre-defined (key) performance indicators PI are given as semi-aggregated values for differentiating analysis, and to enable future investigations which were not foreseen within this evaluation framework. Mainly other researchers and to a certain degree reviewers might be interested in.

Their comparison to the defined base case (conventional method) and their evaluation discussion lead to indications which application is promising for deployment under certain circumstances. Different degrees of equipment with sensors, i.e., the so-called penetration rate, are considered. In conjunction with the business cases defined in the exploitation plan [COLOMBO D6.3, 2015] this is important for all therein named stakeholders, i.e., public authorities and road operators, companies delivering traffic management solutions, road users and inhabitants, traffic engineers and consultants, as well as standardising and regulating bodies.

### 1.3 Work/Document Structure

The structure mirrors the one of section 2.1 in the exploitation plan, where all traffic management solutions are described. These evaluated subsystems can be seen in the following Figure 2 presented as Software-in-the-Loop (SIL), separated from the auxiliary environment simulation components which are not treated in here. They are:

- 1 Traffic state estimation (TSE)
- 2 Automatic incident detection (AID)
- 3 Emission monitoring System (EMS)
- 4 Self-organising traffic control (SOTL)
- 5 Automatic configuration software (tuning)

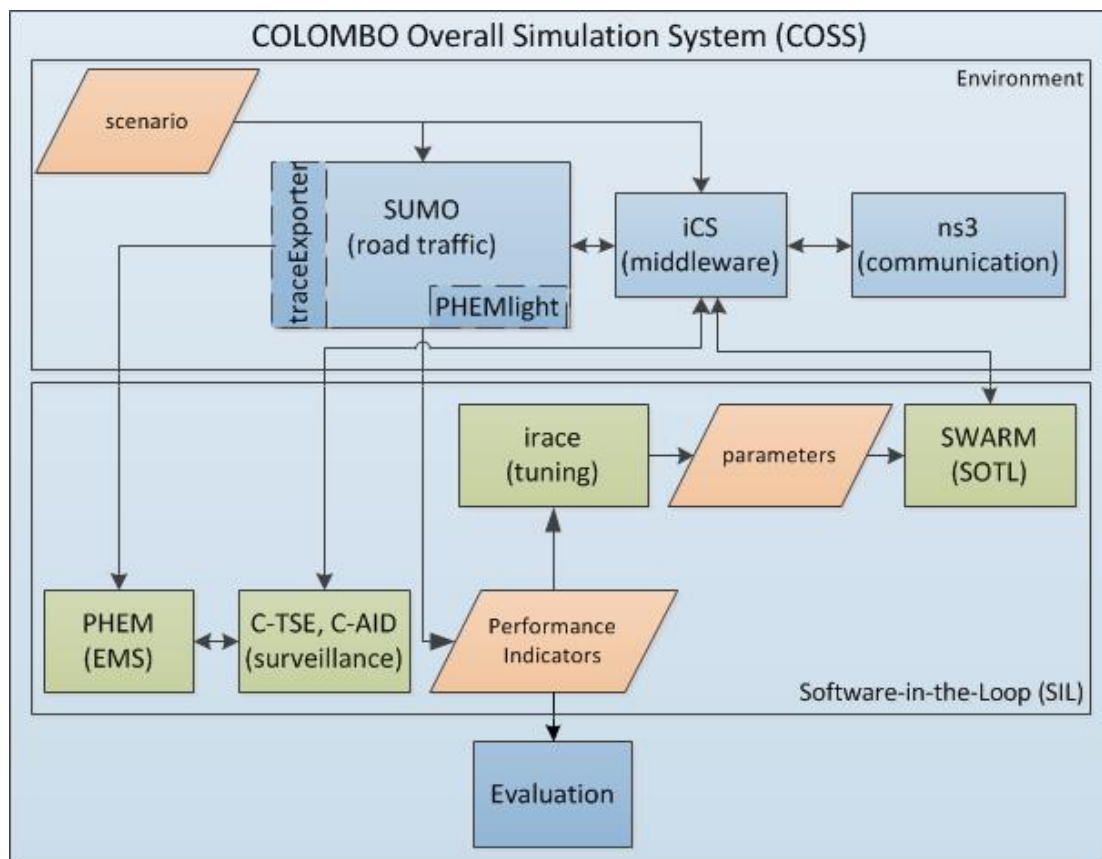


Figure 2: COLOMBO Overall Simulation System (COSS) flow chart

## 2 Traffic state estimation (TSE)

Different algorithms for the cooperative Traffic State Estimation were developed by the project partners, and described in the deliverables [COLOMBO D1.2, 2015] and [COLOMBO D1.3, 2015]. To allow comparability their evaluation took place on the identic scenario set, i.e., the isolated RiLSA junction #1 ([COLOMBO D2.3, 2014], sect. 4.2) with the traffic demand equalling a daily load curve but compressed to one hour (cf. Figure 4). A static traffic light control program is run to avoid result interferences due to the different penetration rates. This scenario parameter “penetration rate” is iterated at 100%, 50%, 25%, 10%, 5%, 2.5%, and 1%. The network is shown in Figure 3.

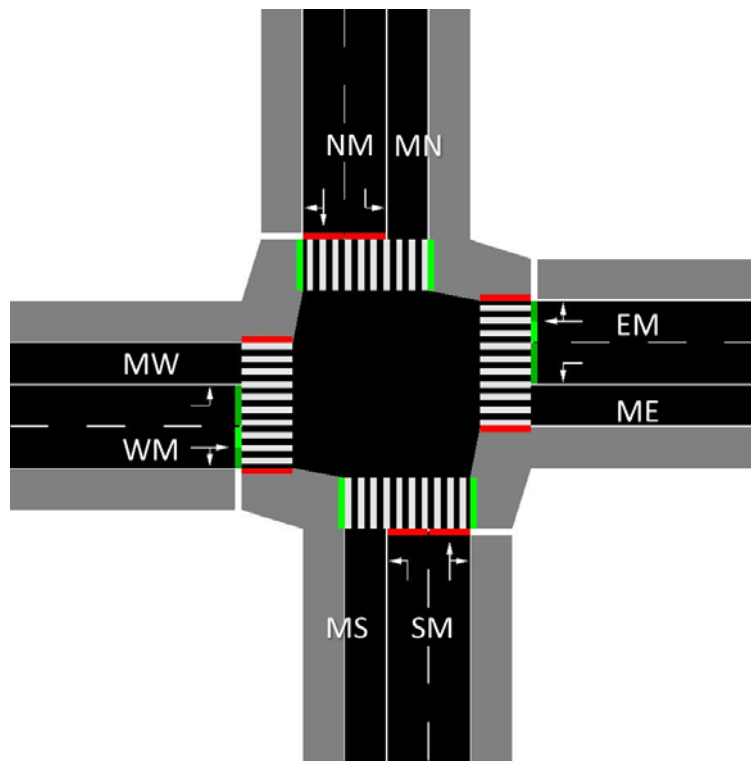


Figure 3: RiLSA junction with edge names.

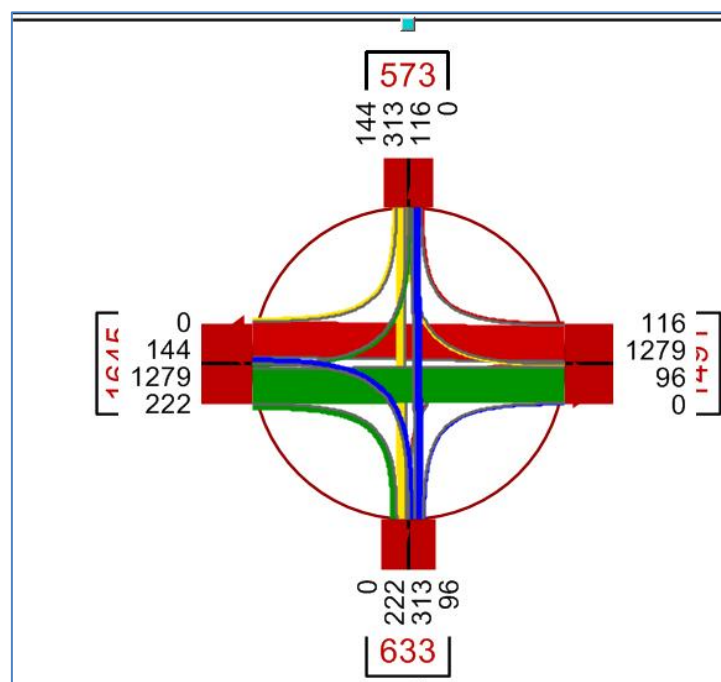


Figure 4: Flows on the RiLSA junction [veh/h]

The evaluation regards two traffic measures for which the Root Mean Square Error (RMSE) between the (simulated) ground truth and the estimated values of the TSE application is determined for each of the four arms separately:

- Mean Speed of vehicles in the detection area and
- Vehicle Count (traffic volume) on a cross section of the road.

The simulator studies were conducted on a software system according to Figure 5.

Cost estimations for initial installation and annual operation are included where possible, taken over from the “Exploitation Plan” [COLOMBO D6.3, 2015]. They can be compared to a four arm intersection equipped with loop detectors which requires an installation cost of 12,000 € and running cost of 1,700€/p.a., as calculated in [COLOMBO D1.2, 2014].

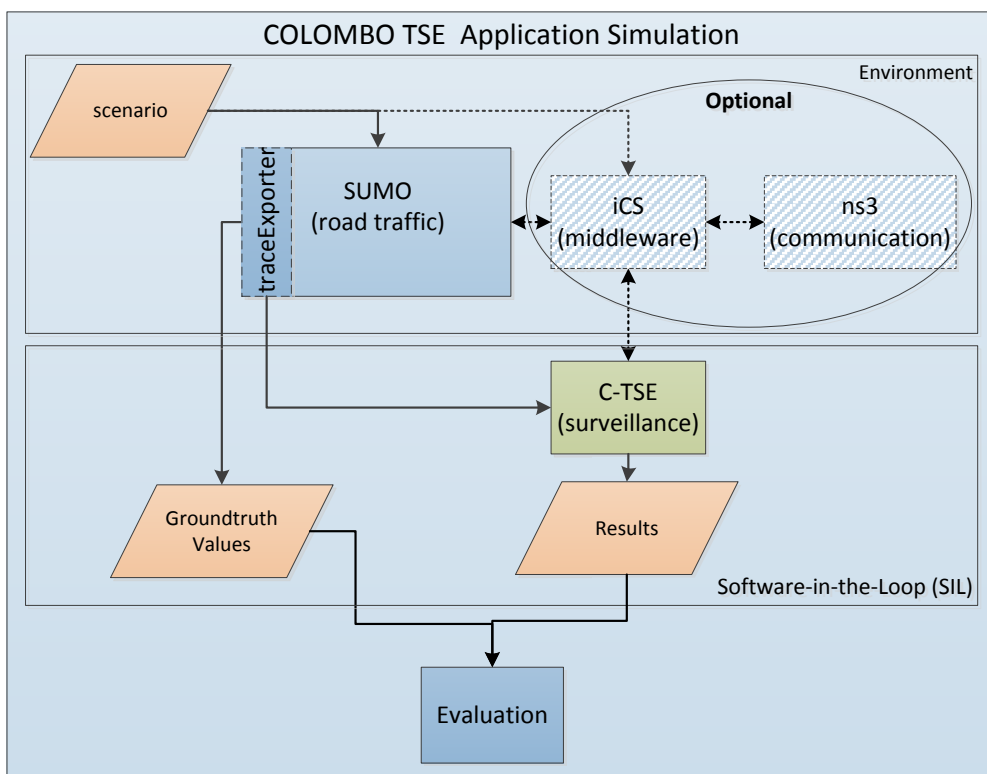


Figure 5: TSE Simulation flow chart

## 2.1 Algorithm 1: Cluster-based

The algorithm was described in [COLOMBO D1.2, 2015] section 4.1.

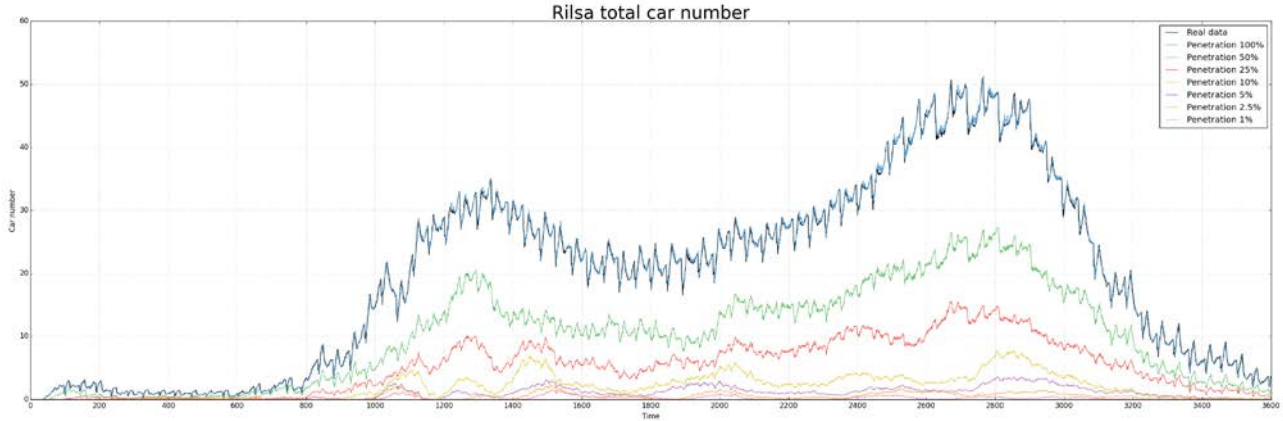
To obtain the results we average 33 different simulation runs. In the charts the black line is the real (ground truth) value, obtained from SUMO without alterations; the coloured lines are the estimations of the protocol for the different penetration rates, as explained in the legends.

Figure 6 shows the total number of cars on every edge of the network for all penetration rates over 1h of simulation. For clarity the other charts show results taken over a smaller time span, which corresponds to the higher traffic peak.

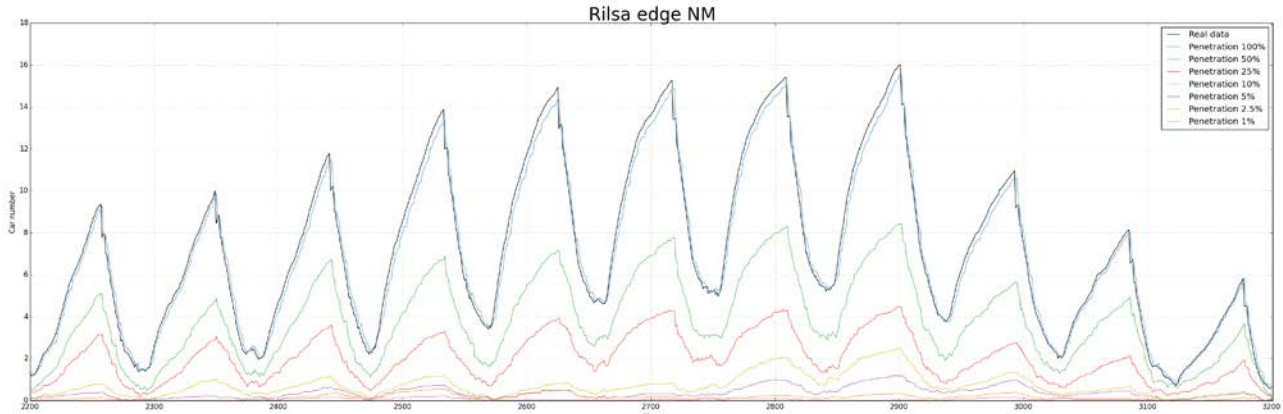
Figure 7 and Figure 8 depict the results on the edges NM and SM respectively. The number of cars estimated by the protocol is approximately directly proportional with the penetration rates: every time the penetration halves the estimation follows the same pattern. Nevertheless, for every



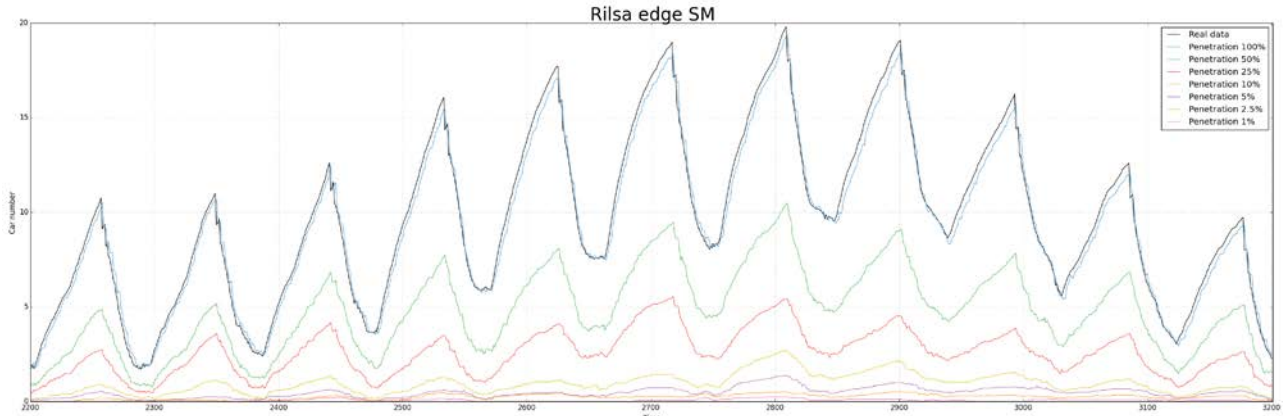
penetration, the estimation follows fairly accurately the shape of the real values, even if the absolute numbers are lower.



**Figure 6: Traffic flow estimation for the different penetration rates.**



**Figure 7: Number of cars in the edge NM of RiLSA.**



**Figure 8: Number of cars in the edge SM of RiLSA.**

Figure 9 and Figure 10 show the average speed of the vehicles estimated by the protocol in the same instants as the previous charts. The black line is again the real average speed obtained from SUMO. As shown in the figures, the average speed is less influenced by the penetration rate, with results that more closely resemble each other. Only for the lower penetration settings there are significant deviations from the mean shape, since the lower number of cars used to get the average worsen the overall result.

The Root Mean Square Errors (RMSE) for both traffic measures are given in Table 1 and Table 2.

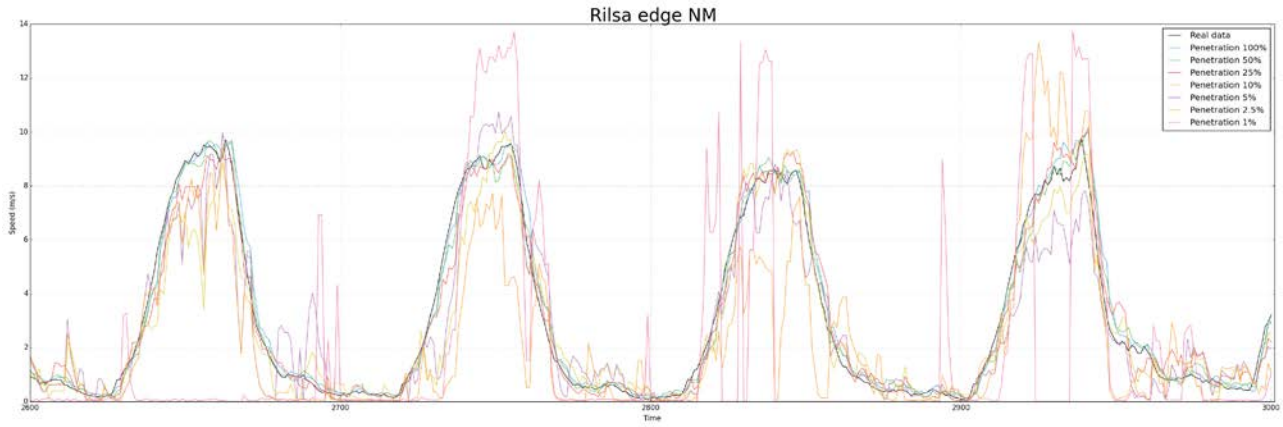


Figure 9: Average speed in the edge NM of RiLSA.

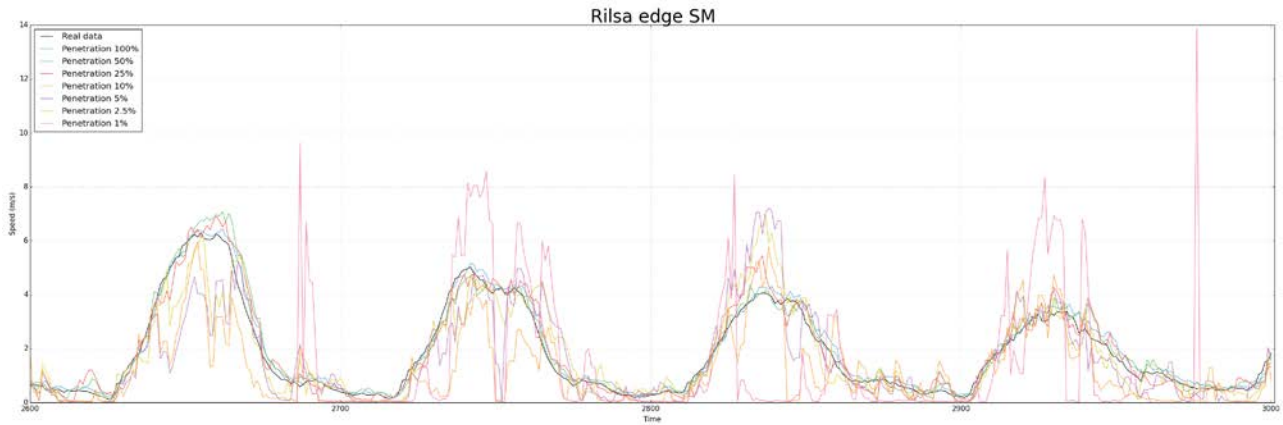


Figure 10: Average speed in the edge SM of RiLSA.

Table 1: RMSE for number of vehicles per edge.

Penetration rate	Total	edge NM	edge SM	edge EM	Edge WM
100%	0.73	0.32	0.36	0.30	0.30
50%	11.85	2.40	3.13	2.46	2.10
25%	17.79	3.62	4.57	3.64	3.06
10%	21.84	4.45	5.57	4.40	3.72
5%	23.16	4.69	5.89	4.66	3.96
2.5%	23.81	4.83	6.05	4.79	4.05
1%	24.17	4.91	6.15	4.86	4.10

Table 2: RMSE for average speeds per edge.

Penetration rate		edge NM	edge SM	edge EM	edge WM
100%		4.55	4.61	4.08	4.34
50%		4.46	4.28	3.93	3.99
25%		3.98	3.85	3.69	3.75
10%		3.59	3.30	3.26	3.24
5%		3.47	3.14	3.30	3.42
2.5%		3.97	3.52	3.73	3.69
1%		3.89	3.62	3.77	3.59

## 2.2 Algorithm 2: DissFlow

The applied DissFlow protocol was described in [COLOMBO D1.2, 2015], section 4.2, and in [COLOMBO D1.3, 2015].

The traffic density estimation can be obtained by these three methods:

- **Delay** – this is the basic method, based on the fundamental relationship between traffic density and communication delay.
- **Neighbour** – this is computed by a moving average of the size of the location table (the number of 1-hop neighbours) by each node relaying a DissFlow packet.
- **Cluster** – this is computed by counting the number of connected vehicles behind the cluster leader reaching the traffic light

Beside RMSE values for V2X-only communication the performance was also investigated for Bluetooth-only, and for a data fusion of both at various combinations of penetration rates. The results are given for the West-East corridor of the RiLSA scenario. The C2X penetration rates of 1.0, 2.5, and 5.0% were not tested.

Figure 11 shows the ground truth and estimated values of the Cluster method with V2X only for the traffic density.

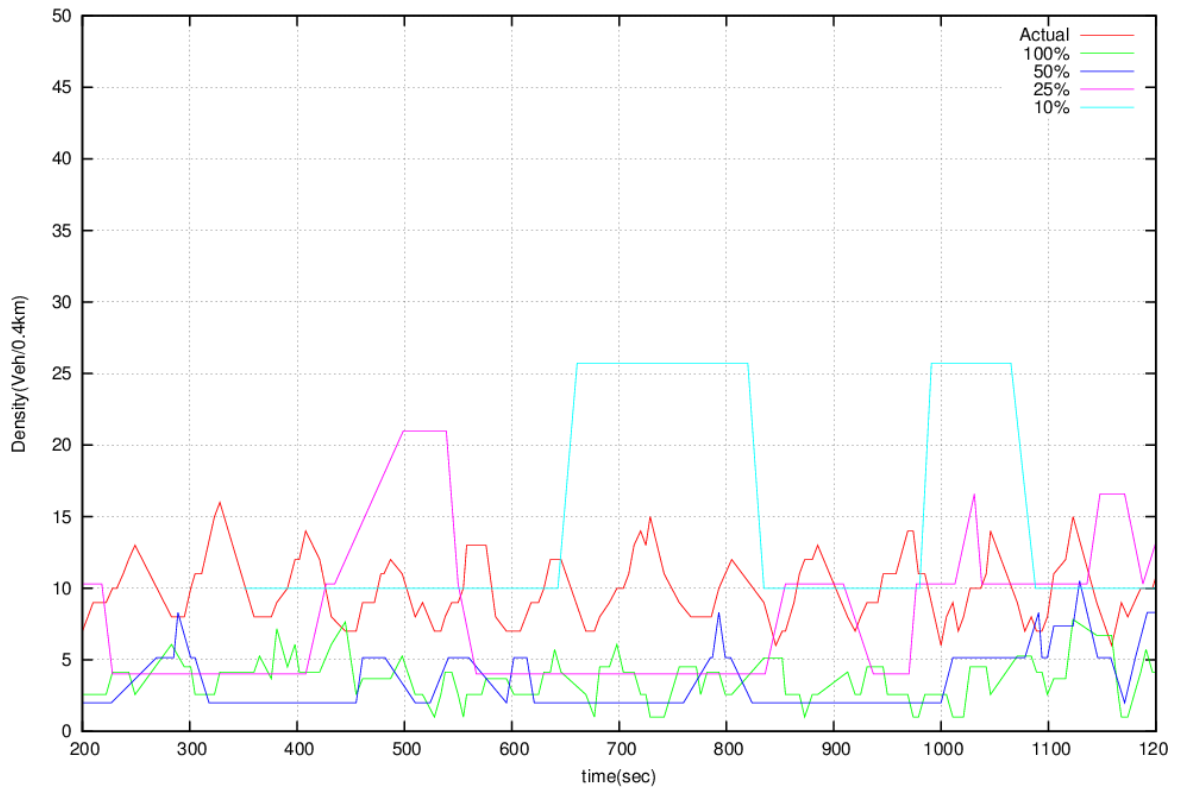


Figure 11: Traffic density of the Cluster method with V2X only

As can be seen in Table 3 the RSME remains high, regardless of the penetration rate or method. The major reason comes from the fact that the traffic lights in RiLSA are influencing the inter-distance hypothesis behind DissFlow, and therefore adding errors to the estimation.

For 100% penetration, we can see that both Delay and Cluster provide the best traffic density estimates. With decreasing penetration, we can see that the Delay method becomes unreliable, while the Cluster method remains stable. Yet, the best performance remains for the Cluster approach. Both Delay and Neighbours are not adapted in the case of RiLSA.

It can be noted that the Cluster approach provides important information for traffic light control, as once a V2X-equipped vehicle reaches a traffic light, it can give the number of vehicles behind it. It can also be noted that the penetration rate has little influence on the RMSE of the Cluster method, making this method quite stable and reliable already at low penetration.

**Table 3: RMSE for density (C2X only)**

Penetration rate	Delay	Cluster	Neighbour	Methods Fusion
100%	6.41	6.84	11.35	3.91
50%	18.98	6.45	16.03	9.78
25%	39.06	6.15	22.28	19.63
10%	61.47	9.00	43.36	35.79

Considering Bluetooth technologies, the precision of the traffic density estimation is different as shown in Table 4. The low transmit range of Bluetooth makes the Delay and Neighbour methods also highly unreliable. In contrast to V2X, the Cluster method does not remain stable with reduction of the penetration rate. The Cluster method nevertheless remains the most precise method of the three, even at low penetration rates.

**Table 4: RMSE for density (Bluetooth only)**

Penetration rate	Delay	Cluster	Neighbour
100%	52.4	5.93	42.5
50%	96.0	10.66	34.21
25%	11.0	16.7	62.84
10%	117.36	26.45	144.95

We observe in Table 5 that among the individual methods, the Cluster method still remains the most precise traffic density estimation, although the impact of the Bluetooth technology is not too beneficial, considering that the RMSE of 10% V2X is approximately the same as for 50% Bluetooth. Finally, a fusion learning algorithm to fuse data between all three methods manages to balance and reduce the RMSE of the worse methods.

**Table 5: RMSE for density (data fusion)**

Penetration rates: DSRC & BLT	Delay	Cluster	Neighbour
50%-50%	41.73	6.15	33.96
25%-25%	84.10	12.7	47.6
10%-10%	99.01	21.3	115.8

The speed estimation shows more stability than the one for traffic density. Considering individual technologies, we can observe that the RMSE of both technologies is reduced by reducing the penetration rate. Although this could be thought as counter-intuitive, it may actually be explained by the fact that with low penetration (or small transmit range), vehicles are more likely not to find good relays and need to keep the DissFlow packets while moving. Accordingly, the speed estimations correspond to the sum of individual estimations. When DissFlow is relayed, the mean speed is computed by a moving average between all relaying nodes, which introduce bias. The drawback is the number of samples, which is lower when DissFlow packets are carried instead of relayed, and the only way to increase the number of samples is to increase the transmit range to get DissFlow packet better relayed, which Bluetooth cannot do.

Comparing average speed estimations for both DSRC and Bluetooth technologies, it can be seen that with a shorter transmit range, the latter provides smaller RSME.

Finally, when fusing both technologies, we can compensate the drawback of a low penetration at high range with a high penetration and short transmit range and reduce the joint RMSE.

Nevertheless, both technologies provide significantly smaller RSME of speed estimation compared to density estimation.

**Table 6: RMSE for speed**

<b>Penetration</b>	<b>100%</b>	<b>50%</b>	<b>25%</b>	<b>10%</b>
DSRC	5.13	4.65	3.65	2.98
Bluetooth	3.35	2.70	2.18	2.31
Fusion (same rates)		3.351	3.35	3.37

This solution could be marketed for an initial cost of 6,000€ and running costs of 400€/p.a. That is 50% resp. 25% of the costs for the loop detector solution.

### 2.3 Algorithms 3 and 4:

Algorithm 3 “Traffic State Generation from CAMs” in [COLOMBO D1.2] was designed for studying performance indicators over longer aggregation intervals rather than generating real time traffic state information for use in a traffic light control application. Therefore, RMSE of averaged speed values determined with this algorithm were not evaluated.

Algorithm 4 “Probe Vehicle Data” in [COLOMBO D1.2] was designed in conjunction with an extended static fixed time TLC and therefore also evaluated only in conjunction with the altered TLC program. RMSE values for this algorithm have not been taken into consideration.

This solution could be marketed for an initial cost of 1,000€ and running costs of 100€/p.a.

### 2.4 Algorithm 5: Data Fusion

A novel approach including data fusion via Bayesian Networks (BN) was described in [COLOMBO D1.3, 2015]. Bluetooth equipment rate was assumed to be 30% throughout all scenarios, with a directed detection range of 15m. C2X penetration rate of 20% (instead of 25%) was used.

In Table 7 the RMSE for the mean speed determination is shown. In case of the eastern and western arms it decreases as expected from 4.5 to 2.3 and from 5.0 to 2.4 m/s, respectively. Due to the higher accuracy of the more frequent C2X speed estimations the maximum of the a-posteriori probability distribution is shifted to more accurate values since the contribution of the less accurate Bluetooth detector decreases.

It should be noted that the RMSE is almost constant (it even slightly increases) for the northern and southern intersection arms for all penetration rates at about 2.2 to 2.4 and 1.8 to 2.1 m/s, respectively. This behaviour can be explained by the fact that the vehicles statistically stick to the lower speeds, which is modelled by the a-priori probability. Therefore, independently on the data the Bluetooth sensor as well as the C2X detector provide, the maximum of the a-posteriori probability distribution provides low speed values, which is statistically true. The slight increase of the RMSE in case of higher C2X penetration rates (although the C2X detector is more accurate than the Bluetooth sensor) is also a resulting effect. In case of very high C2X penetration rates the massive presence of accurate C2X data enables the displacement of the maximum of the a-posteriori probability distribution to increase the overall accuracy again.

Table 7: RMSE for speed

<b>Penetration rate</b>	<b>northern arm NM</b>	<b>eastern arm EM</b>	<b>southern arm SM</b>	<b>western arm WM</b>
100%	1.8	1.8	1.6	1.9
50%	2.2	2.3	2.1	2.4
20%	2.4	3.1	2.1	3.3
10%	2.4	3.5	2.1	3.9
5%	2.3	4.0	1.9	4.4
2%	2.2	4.4	1.9	4.7
1%	2.3	4.5	1.8	5.0

This solution could be marketed for an initial cost of 6,000€ and running costs of 400€/p.a. That is 50% resp. 25% of the costs for the loop detector solution.



### 3 Automatic incident detection (AID)

The approach of the applied algorithm “Probability Density Maps” (PDM) is to gather statistics of various road users’ behavior parameters. There are numerous parameters that can be used to assess if the behavior of a road user is as usual or not. Reasonable parameters in this regard are location-based vehicle count, vehicle velocity, the heading angle of the vehicle and many more. The idea of a PDM is to link the information on the expected value of these parameters to the location within the scene. A given trajectory can then be assessed with respect of its likeliness to appear in a road scene.

Two experiments have been conducted according to the flow chart in Figure 12: The first one represents the best case regarding the quality of C2X data (cf. Figure 14). The errors comply with the errors of the video-based detection system, having a mean position error of 2.36m, a median position error of 1.70m, and a standard deviation of 2.46m. The second one is the worst case to be expected for the quality of C2X data and has additional noise caused by GPS signal distortion and therefore additional errors compared to the best case scenario.

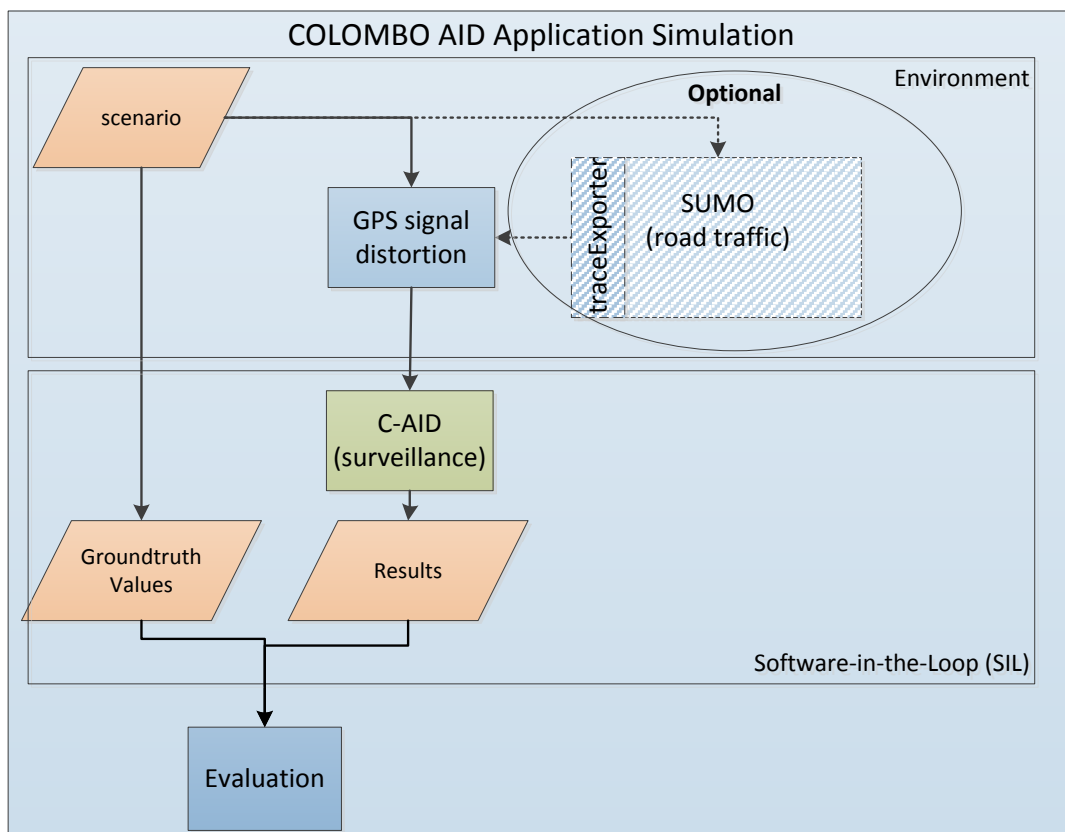


Figure 12: AID Simulation flow chart

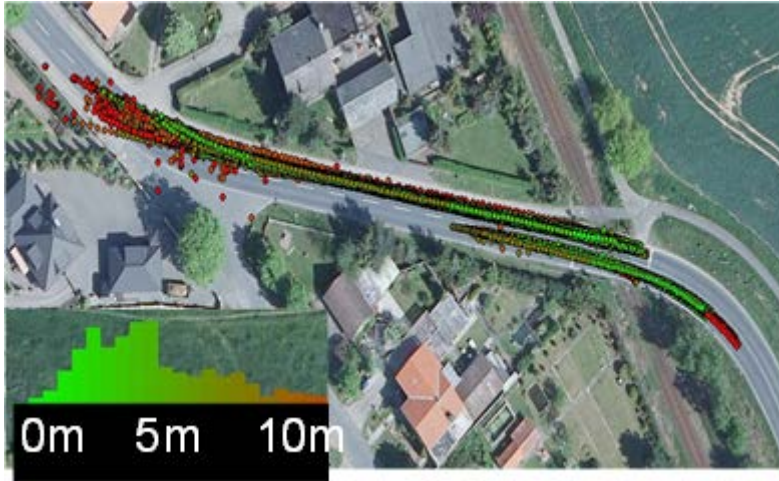


Figure 13: Position errors of best case scenario C2X data

The results of the incident detection experiments are shown in Table 8 to Table 11. As is usual for a classifier, the results of the PDM algorithm are given in a confusion matrix for the two experiment setups and different penetration rates for C2X.

The tables indicate the following:

- The PDM algorithm can deal well with distorted data (scenario 2)
- The higher the penetration rate, the better the detection rate of incidents
- Remarkable is the low false-positive-rate for all penetration rates and in both scenarios, as well as the constant true-negative rate. The very low false-positive-rate indicates there are almost no false alarms to be expected by using PDM
- Per incident: already a penetration rate of 20% allows to detect 50% of the incidents<sup>3</sup>

Table 8: Confusion matrix values for different penetration rates and undistorted trajectories [per frame]

	penetration rate						
	1	0.5	0.2	0.1	0.05	0.02	0.01
<b>TP</b>	41.0%	23.1%	10.8%	4.8%	3.9%	0.2%	0.6%
<b>FP</b>	5.5%	4.0%	2.2%	0.8%	0.7%	0.2%	0.1%
<b>TN</b>	59.7%	59.7%	59.8%	59.8%	59.8%	59.9%	59.9%
<b>FN</b>	0.8%	1.4%	1.8%	2.0%	2.0%	2.2%	2.2%

Table 9: Confusion matrix values for different penetration rates and GPS noise added [per frame]

	penetration rate						
	1	0.5	0.2	0.1	0.05	0.02	0.01
<b>TP</b>	41.0%	23.0%	11.7%	4.8%	1.3%	2.5%	0.0%
<b>FP</b>	5.6%	3.6%	2.4%	1.4%	0.4%	0.5%	0.0%
<b>TN</b>	59.3%	59.3%	59.4%	59.4%	59.4%	59.4%	59.5%
<b>FN</b>	0.8%	1.4%	1.8%	2.0%	2.1%	2.1%	2.2%

<sup>3</sup> These results should be treated with caution, since few incidents happened within the time interval.



**Table 10: Detection rates for incidents at different V2X equipment rates for undistorted trajectories**

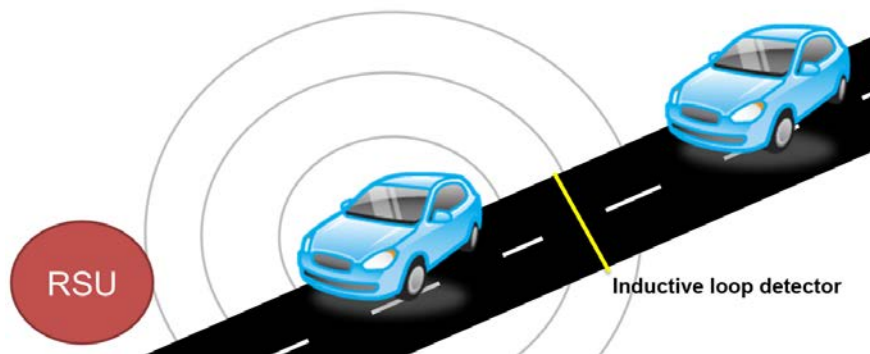
	penetration rate						
	1	0.5	0.2	0.1	0.05	0.02	0.01
<b>TP</b>	100%	84%	48%	22%	14%	4%	4%
<b>FP</b>	0%	0%	0%	0%	0%	0%	0%
<b>TN</b>	100%	100%	100%	100%	100%	100%	100%
<b>FN</b>	0%	0%	0%	0%	0%	0%	0%

**Table 11: Detection rates for incidents at different V2X equipment rates for trajectories with GPS noise added**

	penetration rate						
	1	0.5	0.2	0.1	0.05	0.02	0.01
<b>TP</b>	100%	80%	48%	26%	8%	10%	0%
<b>FP</b>	0%	0%	0%	0%	0%	0%	0%
<b>TN</b>	100%	100%	100%	100%	100%	100%	100%
<b>FN</b>	0%	0%	0%	0%	0%	0%	0%

## 4 Emission monitoring System (EMS)

The Emission monitoring System was described in detail in [COLOMBO D1.3, 2015]. This section gives a short summary of the approach of the EMS and the results. The basic idea for emission monitoring is to use the V2X data of equipped vehicles and additionally the measurements from conventional traffic detectors (see Figure 14). In the early adoption phase of V2X communication there will probably be only a low number of vehicles with V2X communication equipment. Therefore, it will be difficult to estimate the total emissions of an intersection by taking the data from V2X communication only into account. Additionally, the count and optionally vehicle classification data from inductive loops can help to get a picture of the whole traffic situation. While inductive loops are widely used for calculating traffic flows and adapting traffic light phases, inductive loops alone cannot be used for emissions estimations (because they only measure the number of vehicles at one location). For the emission calculation only one intersection will be monitored. For further research the basic approach can be extended to monitor larger regions as well.



**Figure 14: Data collection via a road side unit and an inductive loop detector**

The Emission Monitoring System (EMS) was implemented with an application which uses SUMO for the traffic simulation and the PHEMlight for the Emission simulation. Alternatively, the produced emissions could be calculated via the traceExporter within SUMO and PHEM. The ground truth for the emissions is estimated via the calculated emissions of all vehicles within the simulation scenario. The EMS application uses only the trajectory data of the vehicles from the simulation for the equipped vehicles only. Afterwards PHEM/PHEMlight is used to calculate the emissions from the trajectory data. The calculated results are compared to the ground truth in the evaluation process (see Figure 15).

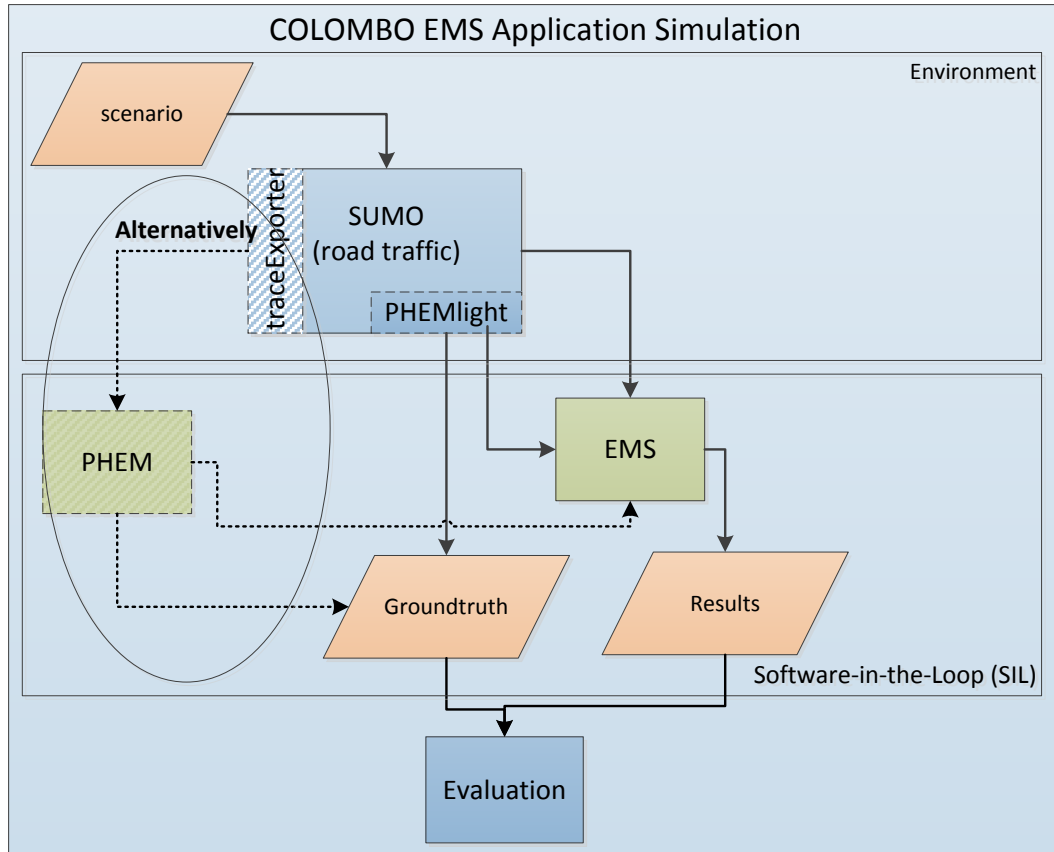


Figure 15: EMS Application flow chart

## 4.1 Basic Approach

The simulation results show that even with a low penetration rate of 1% the relative error of the estimated emissions is around 5% after one hour of simulation and emission collecting time (see Figure 16). The relative error is decreasing with a higher penetration rate. This effect is as expected because with a higher penetration rate the knowledge about the produced emissions is also larger.

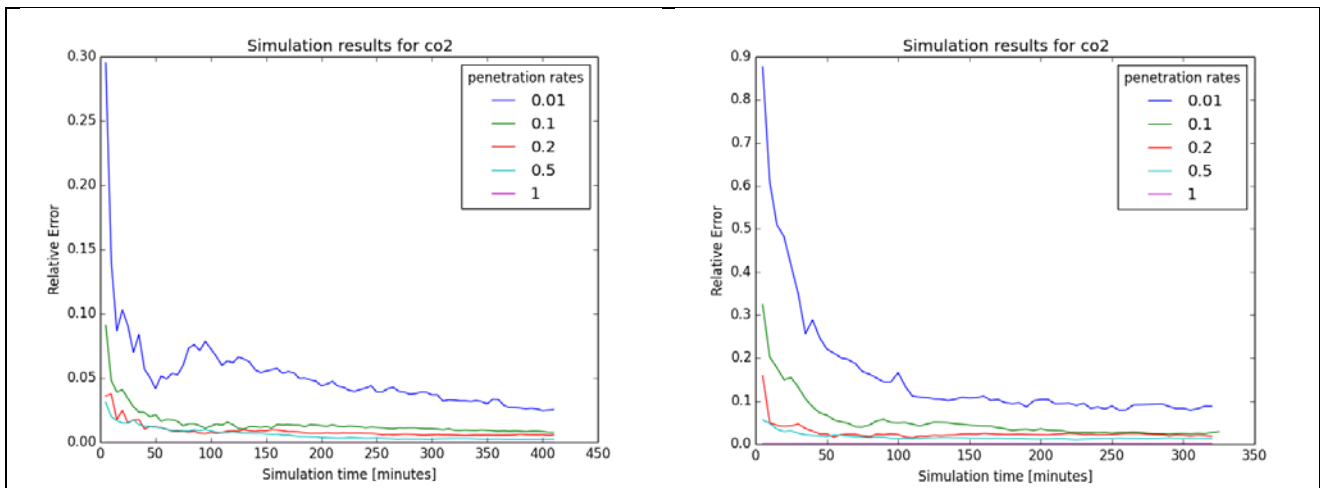
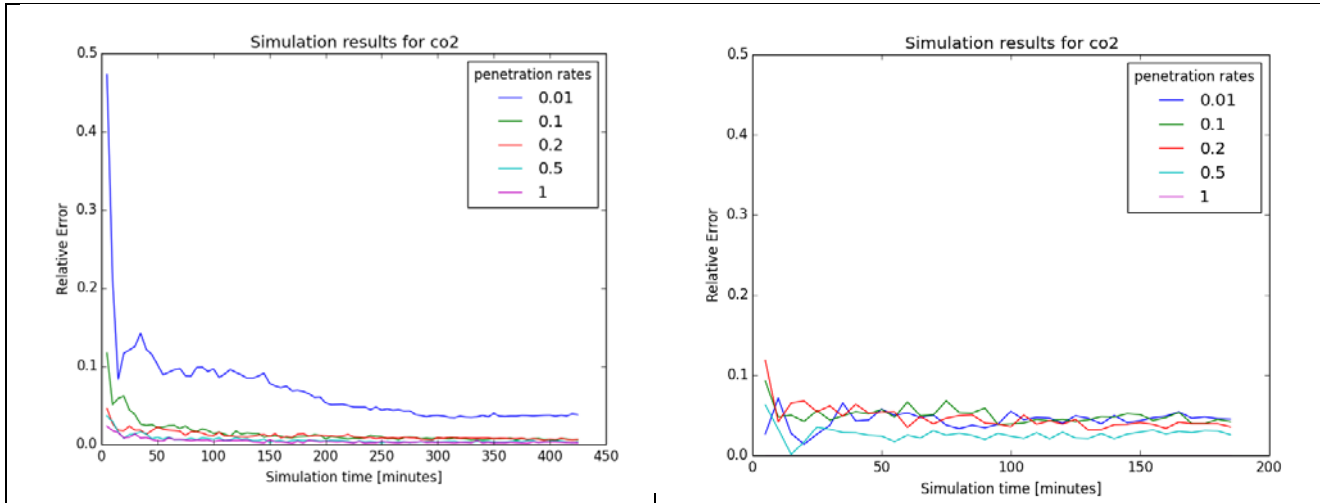


Figure 16: Relative Error of the estimated emissions over time (left). With different vehicle types (right)

## 4.2 Clustering Approach

For the emission monitoring the emission clusters are mapped onto the induction loops or street of the network. According to the assumption that the GPS positions of the CAMs are not accurate enough to know the exact lane and loop a vehicle is driving at, this mapping distinguishes only between the approaching streets, but not the lanes. The errors for the emission calculation with the

clustering approach can be seen in Figure 17. The errors are slightly higher than the results for the basic approach but are still relatively good. When comparing the result for the simulation with different vehicle types the clustering approach performs much better than the basic approach.



**Figure 17: Relative Error of the estimated emissions over time for the clustering algorithm (left). With different vehicle types (right)**

### 4.3 Conclusions

A combined approach to monitor emissions was presented and evaluated within COLOMBO. The basic idea of the approach was to use the vehicle counts from induction loops and combine this information with the speed profile of equipped vehicles. The simulation results show that the algorithm can be used for emissions monitoring. Even with low penetration rates of 1% an error of less than 10% is achieved in the simulations.

But the current state of research has some limitations. The simulation scenario which was used for the evaluation is an artificial one. To have a more realistic traffic situation the approach should also be simulated for an intersection with real world data (an existing intersection and its traffic demand). For this evaluation only simulation results have been considered. Simulation models are normally a simplification of a real world situation. Therefore it is possible that the simulated emissions will have a larger error when comparing to measured emissions in real world. Hence, the results of the algorithm should also be compared to real world measurements. A first step for validating the simulation results is to use real world measurements of trajectory data, as has been done in the investigations related to the COLOMBO Automated Incident Detection system (see section 2).

The described emissions monitoring approaches were evaluated for monitoring a single intersection. Further research should be done to extend the approach for larger regions. In this context, ideas exist to use commercially available FCData (cp. COLOMBO Traffic State Estimation algorithm 4 in [COLOMBO D1.2]) or to collect CAMs inside the vehicle over a longer time and sending it via GSM or road side units.

## 5 Self-organising traffic light control (SOTL)

The following description follows the standardised reporting for simulation of traffic signal control systems [ISO, 2015].

### 5.1 Simulation Conditions

#### 5.1.1 Simulator

As for all project investigations the open source microscopic traffic simulator SUMO was applied in its version 0.24.

Simulation guidelines from FGSV and FHWA were partially applied. Model validation was not possible due to a lack of an independent reference data set.

Each scenario was run 30 times and the monitored performance indicators averaged by a separate script.

#### 5.1.2 Road Network

To compare the developed SOTL algorithm called SWARM with a state-of-the-art adaptive TLC implemented in a realistic environment, a new network scenario had to be introduced. The existing real-world scenarios did not apply adaptive TLC.

The northern Italian city of Monza was interested to implement the already commercially available adaptive TLC algorithm “ImFlow” of the COLOMBO project partner Imtech. Based on partially disclosed information [CST, 2012], a stretch of three frequently congested signalised intersections (I1, I3, I4) on the south-western Viale Sicilia was modelled and investigated (Figure 18). I2 is a yielding junction, and I5 is a complex roundabout which serves only as origin/destination. The distances of the links are contained in Figure 18.

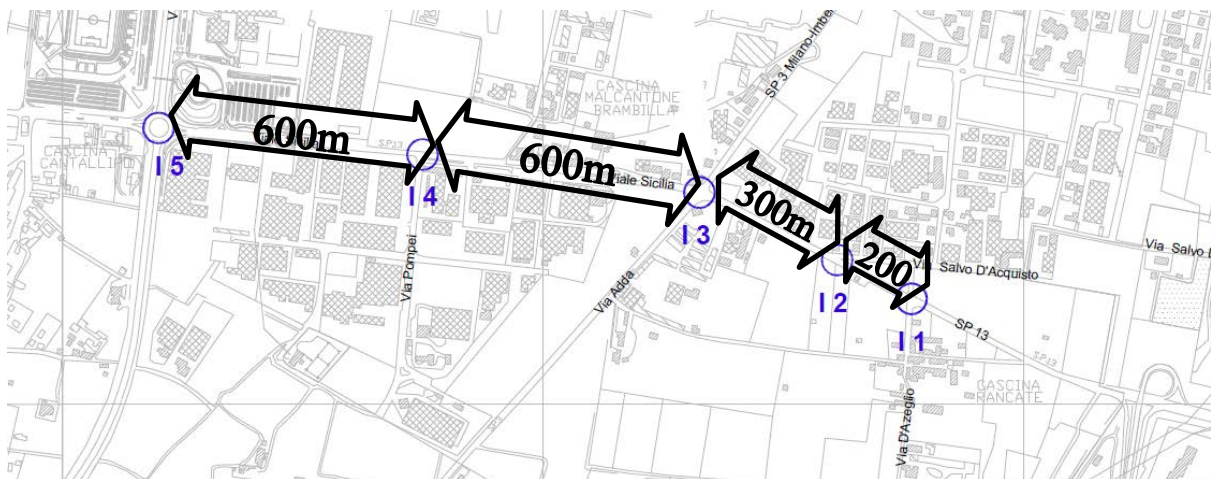


Figure 18: Road Network Configuration

Figure 19 to Figure 21 show the number of lanes, their directions in the proximity of the junction, and for fixed time control the timings of stages and interstages / phases (cf. section 5.1.5). Connecting links in-between have single lanes which broaden to two lanes approximately 150m before and after each signalised junction.

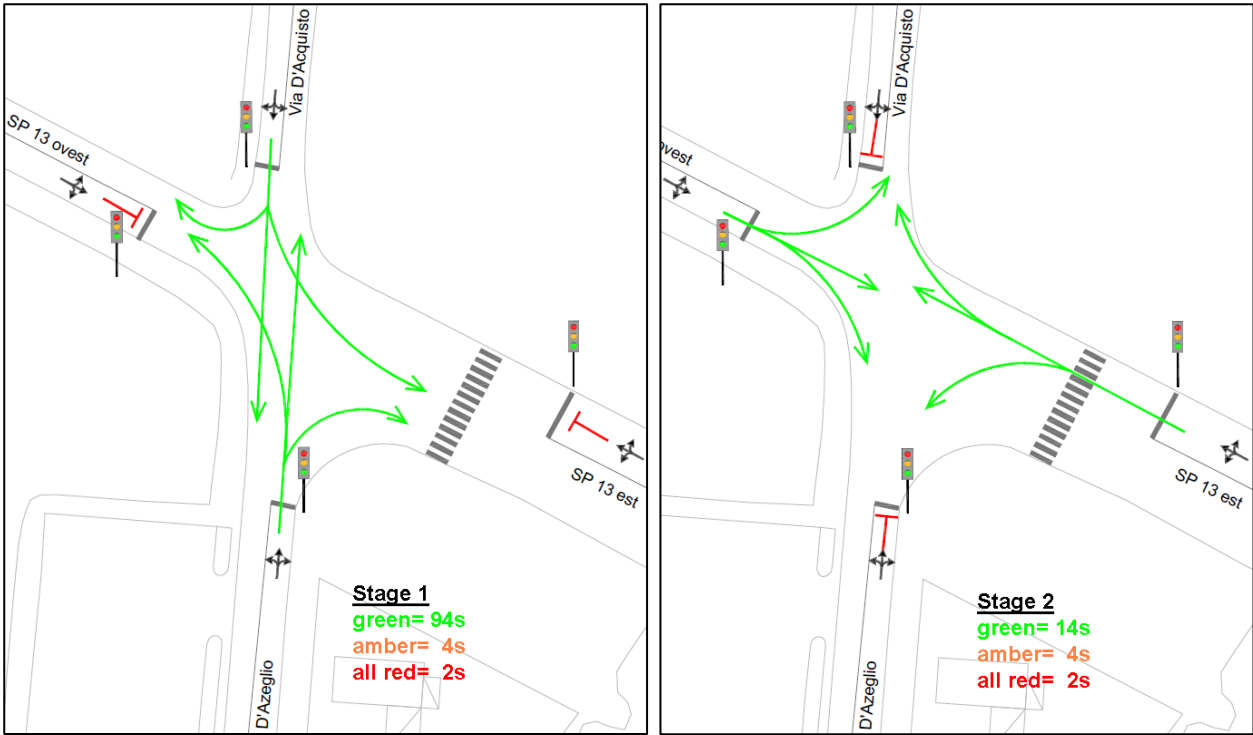


Figure 19: Lanes, layout, and cycle stages (own calculation) at Monza junction I1 [CST, 2012]

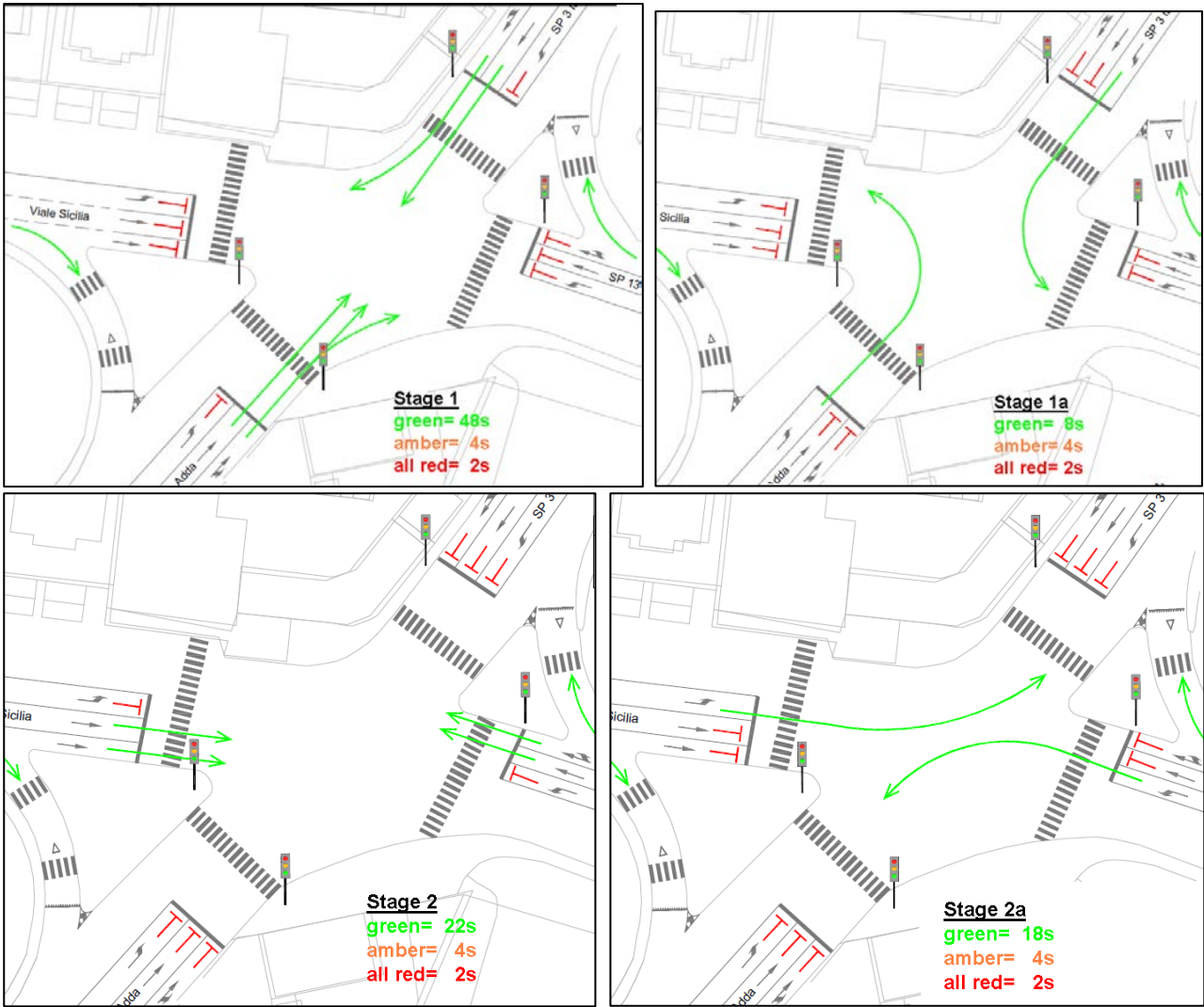


Figure 20: Lanes, layout, and cycle stages (own calculation) at Monza junction I3 [CST, 2012]





Figure 21: Lanes, layout, and cycle stages (own calculation) at Monza junction I4 [CST, 2012]

### 5.1.3 Detectors

Traditional detection of vehicle counts and speed was modelled with SUMO detectors. Loops of 1m length are placed 1m away from the stopping line in each approaching lane. Gap detectors are 25m long, starting at 10m distance from the stop line. Entry detectors of 1m length are placed right after the uncontrolled intersection I2 and straight at the entrances of the network.

Cooperative detection was emulated as SUMO detectors, too, on all incoming and outgoing lanes around the controlled intersections. The simulator class is named MSSOTLE2Sensors within SUMO.

The penetration rate with C2X equipment is an exogenous scenario parameter and iterated through 100%, 50%, 25%, 10%, 5%, 2.5%, and 1%.

### 5.1.4 Traffic Conditions and simulation time

- Types of demand: 3 types of passenger cars (gasoline EU4, diesel EU4) with a fleet share of each 33%. Additionally on the main road trucks (diesel EU4) have a 10% share of the whole demand. These are assumptions since no detailed counts were available.
- Public bus line Z206 traverses I5 and I4 with very few courses per day. It was omitted.
- Pedestrians were modelled at junctions I1 and I3 according to the provided crossings with an assumed volume  $q=120/h$  in each direction to mirror the arising vehicle turning restraint.

- d) The traffic demand at each junction including all turnings is given in Figure 22 to Figure 24 for the morning peak hour between 7:30 and 8:30 [veh/h]. The report contains all values in 15-min-intervals between 17:00 and 19:00, too, which were transformed into the simulation parameter “probability”. The off-peak demand was set to 60% of the respective peak interval.



Figure 22: Morning peak volumes [veh/h] for all turnings at Monza junction I1 [CST, 2012]

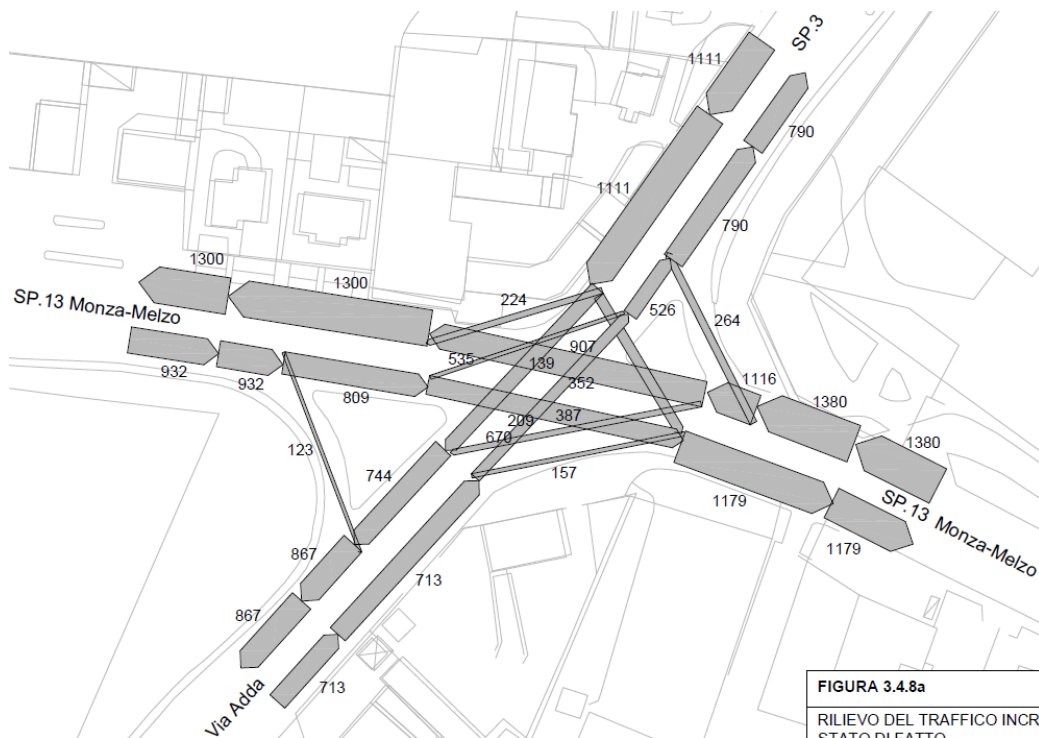
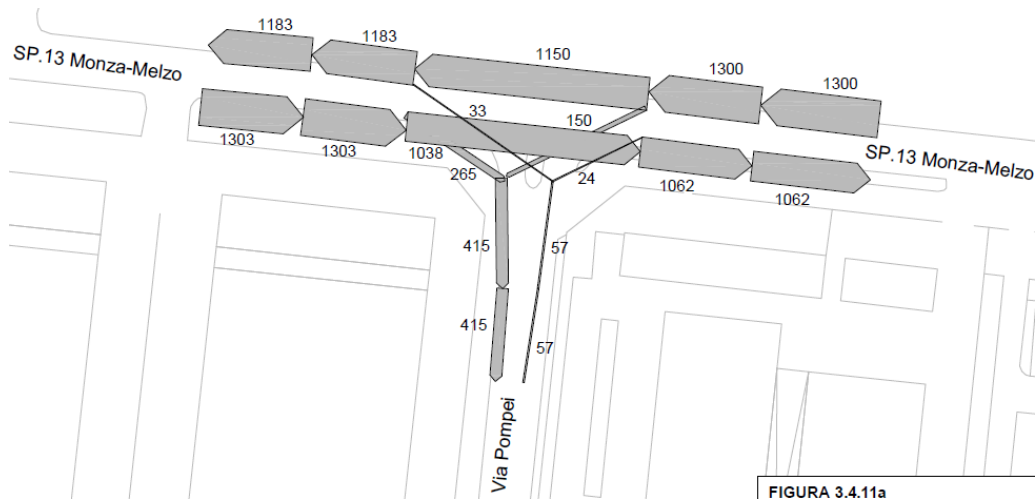


Figure 23: Morning peak volumes [veh/h] for all turnings at Monza junction I3 [CST, 2012]





**Figure 24: Morning peak volumes [veh/h] for all turnings at Monza junction I4 [CST, 2012]**

- e) Distribution of incoming traffic: Poisson (SUMO parameter: probability)
- f) Assumed saturation flow: 2,400veh/h/lane; during peak time the green split for the fixed time control as in Figure 19 to Figure 21 yields in saturation rates below capacity
- g) Maximum speed: 50 km/ (13.89 m/s) in the network; some vehicle classes have higher maximum speed tags but do not exceed the network speed limit
- h) Simulation time: 7,200s (120min / 2h)  
Simulation resolution: 1s  
Pre-simulation time (warm-up): 900s [results of the first interval are omitted]

### 5.1.5 Signal Control

The traffic light control (TLC) to be investigated is the developed SWARM algorithm, described in [COLOMBO D1.3, 2015] and [COLOMBO D2.1, 2014] to [COLOMBO D2.4, 2015]. Flow chart descriptions of the control logic are in Appendix B. SWARM was tuned to minimize the performance indicator (PI) *waiting time*.

In reality the static fixed time TLCs applied in Monza had all different cycle times and were not coordinated. To have a fair base case optimised static fixed time TLCs with a common cycle time of 120s were computed (cf. Figure 19, Figure 20, and Figure 21). To compare the SWARM with more advanced commercially available algorithms an actuated TLC and IMTECH's adaptive ImFlow algorithm [Van Vliet, Turksma, 2013] were modelled, too. ImFlow regards the PIs *waiting time* and *stops* in its objective function.

### 5.1.6 Simulation System Configuration

The SWARM as SOTL control algorithm was implemented into the simulation as shown in Figure 25. The optional simulation of communication with ns3 was neglected due to the high calculation cost which raised performance bottlenecks within this study.

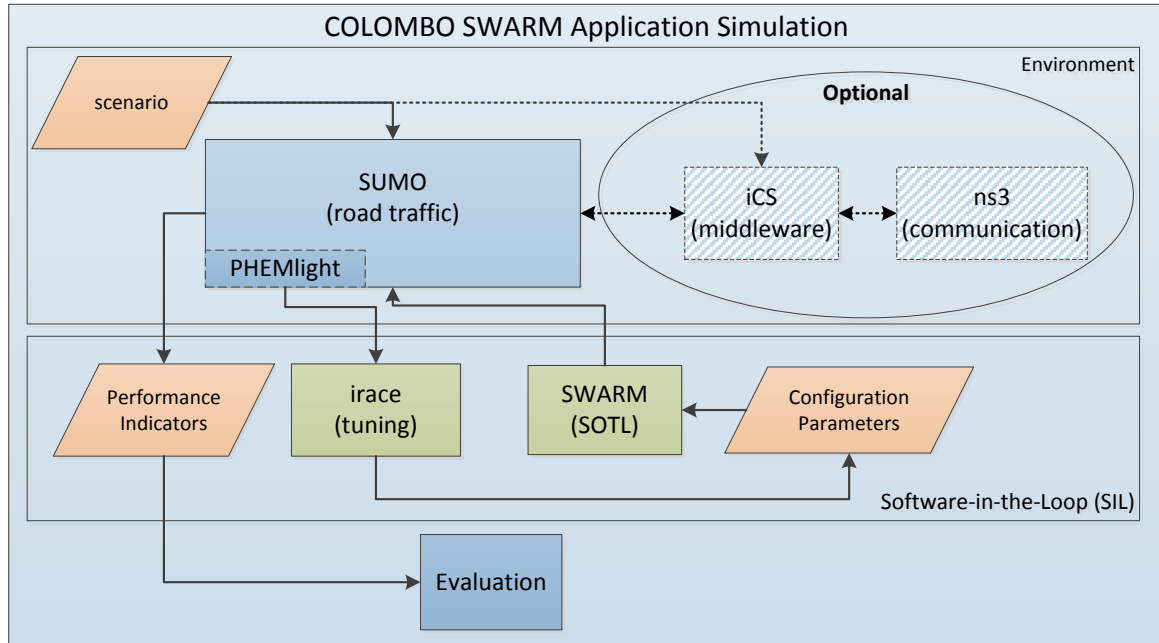


Figure 25: SOTL simulation flow chart

## 5.2 Simulation results

The intended evaluation method and procedure was described in [COLOMBO D5.3, 2014], section 4 and 5. However, the therein developed grade as single evaluation measure could not be applied in the moment of issuing this report. The necessary CO<sub>2</sub> emissions were simulated in an unrealistically high manner due to inappropriate car-following modelling parametrization in SUMO (acceleration noise of Krauss car following model).

As a substitute the PIs “number of stops” and “waiting time” were taken, with their relative change compared to the base case “actuated TLC”. They are shown per junction for both traffic volume scenarios: off-peak on the left, peak on the right side.

For junction I1 Figure 26 with the base case shows that it is as good as the adaptive ImFlow in Figure 27. The SWARM at 1% in Figure 28 has lower waiting time but more stops than ImFlow and the base case. Results for the different penetration rates further improve up to the values for SWARM 50% sketched in Figure 29 and given in Table 12. SWARM outperforms ImFlow not only at the waiting times but sometimes also at the number of stops (SWARM 100% further improves at the number of stops during off-peak, but worsens in regard to all other values).

Figure 26: PIs for base case (fixed time TLC) at Monza I1

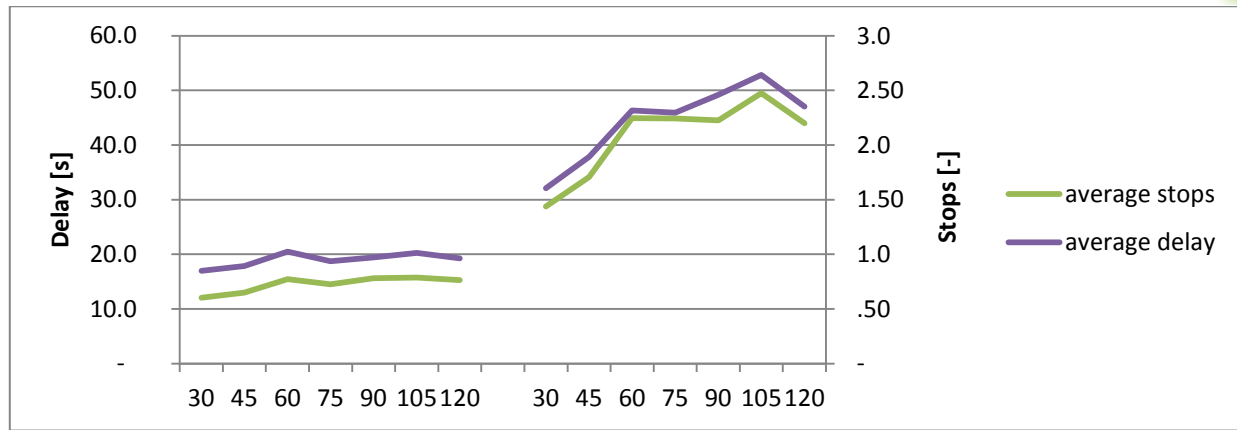


Figure 27: PIs for adaptive TLC at Monza I1

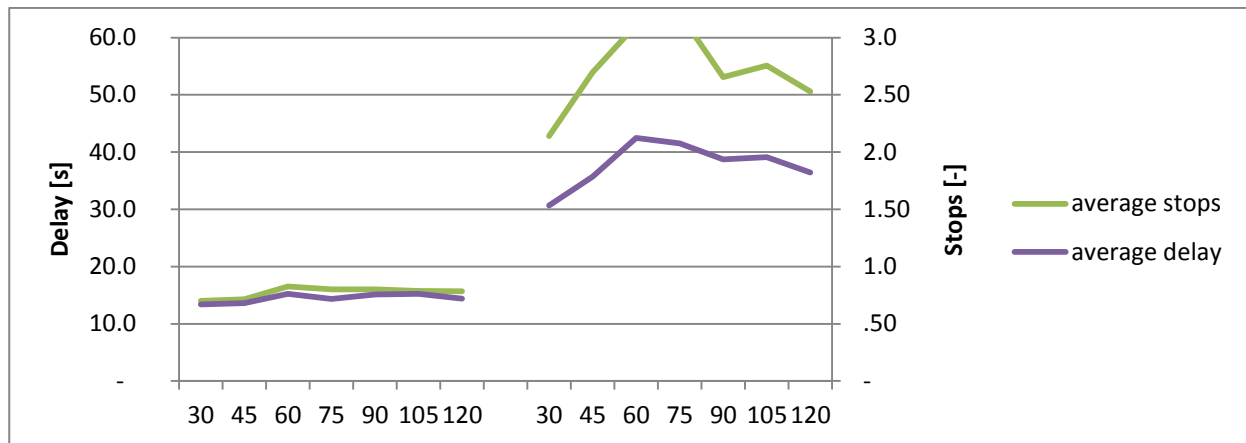


Figure 28: PIs for SWARM TLC 1% at Monza I1

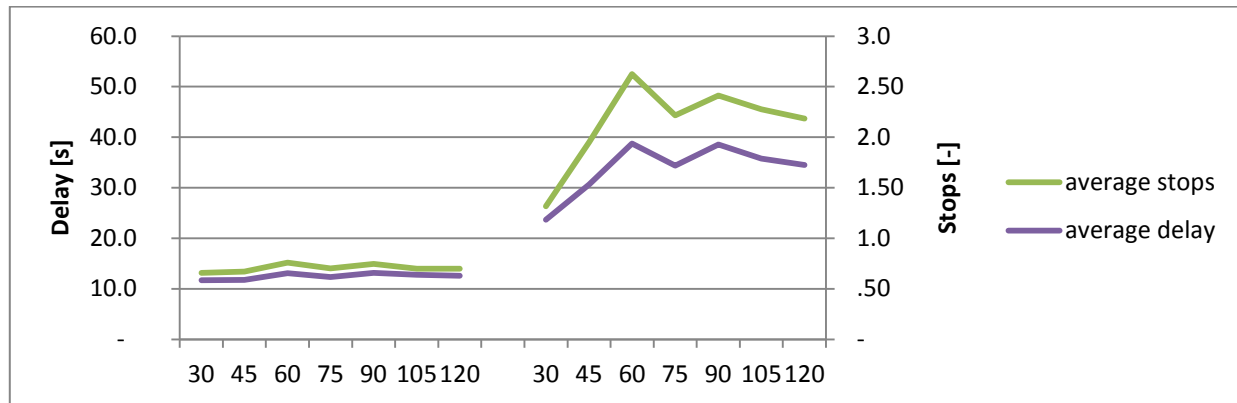


Figure 29: PIs for SWARM TLC 50% at Monza I1

Table 12: Relative PI changes of SWARM 50% against adaptive TLC at Monza I1

Time interval [min]	Off-peak		Peak	
	Stops [-]	Delay [s/veh]	Stops [-]	Delay [s/veh]
0-120	-3%	-34%	1%	-25%
15-30	9%	-31%	-8%	-26%
31-45	3%	-34%	14%	-19%
46-60	-1%	-36%	17%	-16%
61-75	-3%	-34%	-1%	-25%
76-90	-4%	-32%	8%	-22%
91-105	-11%	-37%	-8%	-32%
106-120	-8%	-35%	-1%	-27%

The improvement throughout the different penetration rates as claimed above is exemplary shown for junction I1 in Figure 30. The general tendency of decreasing PI values up to the penetration rate of 50% does not exclude intermediate worsening. Also the values at 100% suggest a backlash from the 50% penetration rate onwards. It must be noted that this interval is quite big and without any further sampling point in between to pin down where this contra-movement starts.

The other two junctions I3 and I4 show similar trends, hence in the following only SWARM results for 1% and 50% will be given.

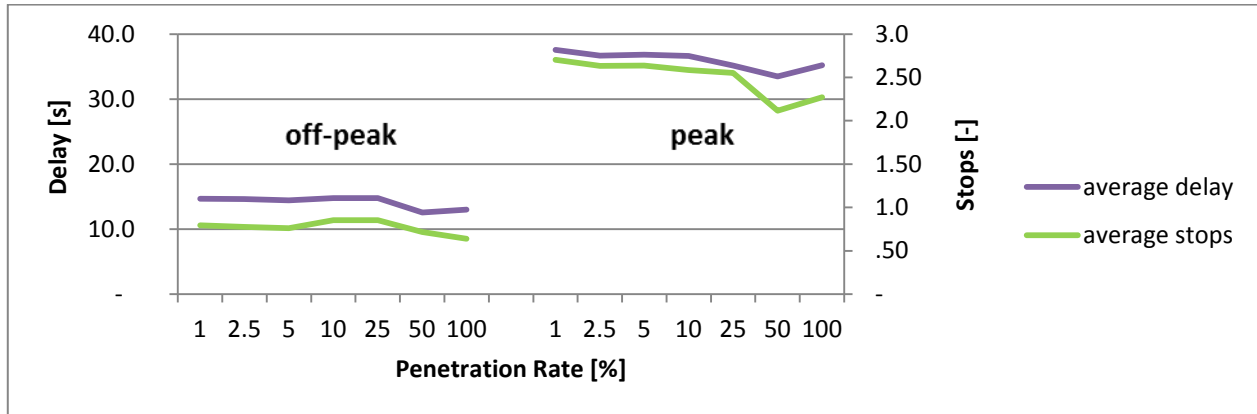


Figure 30: Interval-aggregated PIs for SWARM TLC with all penetration rates at Monza I1

At T-junction I4 the base case (Figure 31) and adaptive ImFlow (Figure 32) have very similar values for both PIs, like at I1. No SWARM penetration rate between 1% (in Figure 33) and 100% can underscore these adaptive/fixed time values, despite SWARM 50% (in Figure 34). At this penetration rate both PIs have slightly better values during most peak time intervals (Table 13).

Generally speaking both PIs are much higher than on I1 during peak period, while during off-peak I4 has more desirable PI values.

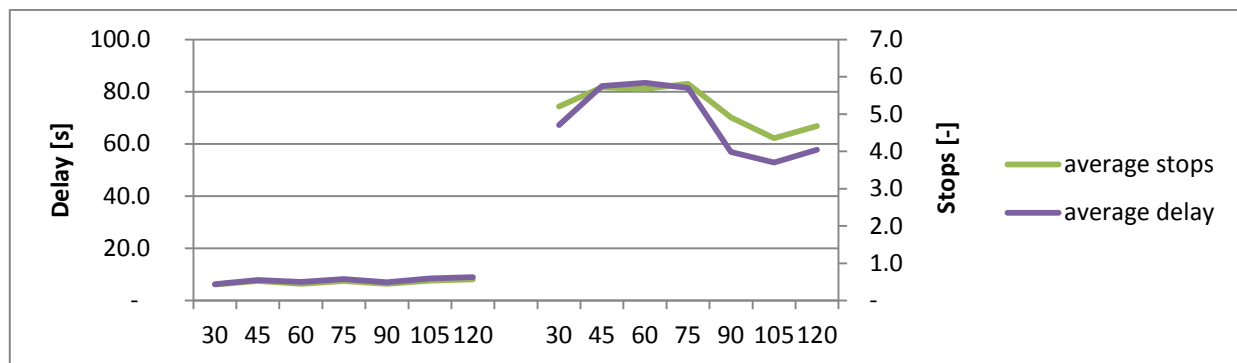


Figure 31: PIs for base case (fixed time TLC) at Monza I4

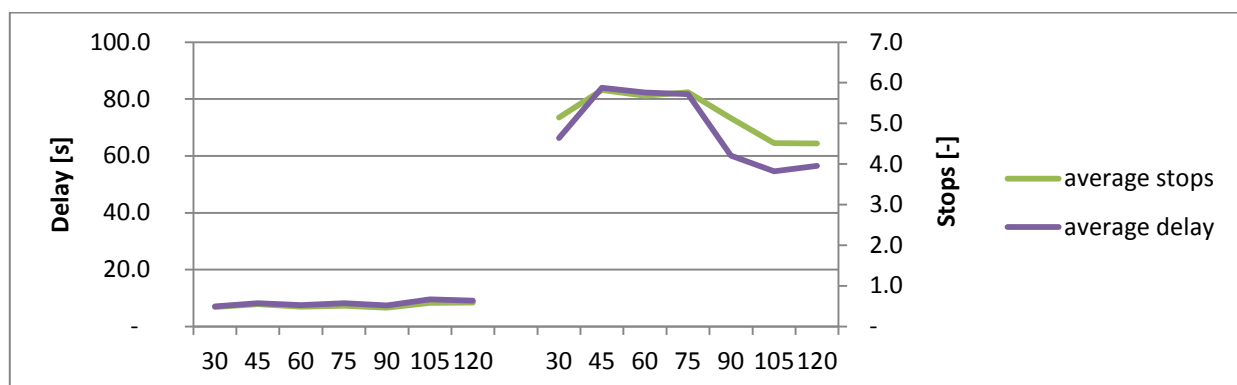


Figure 32: PIs for adaptive TLC at Monza I4

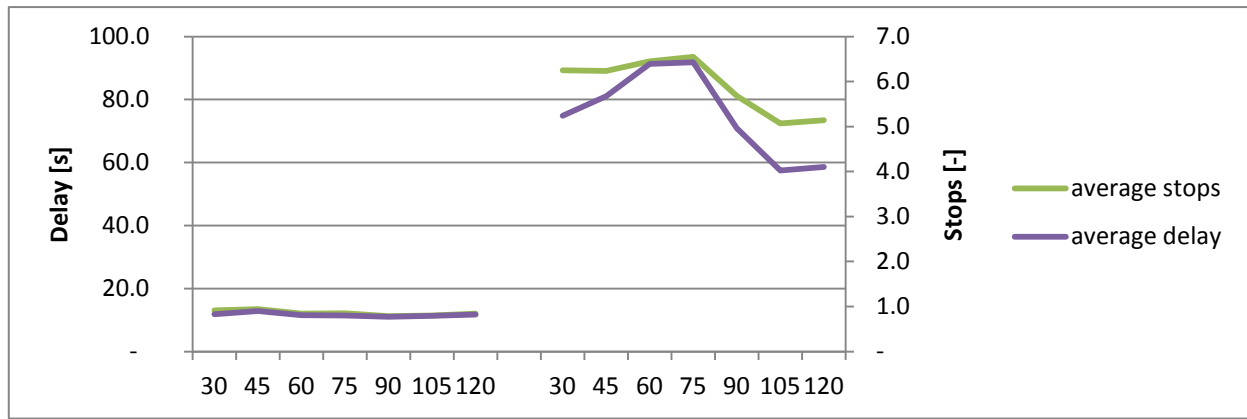


Figure 33: PIs for SWARM TLC 1% at Monza I4

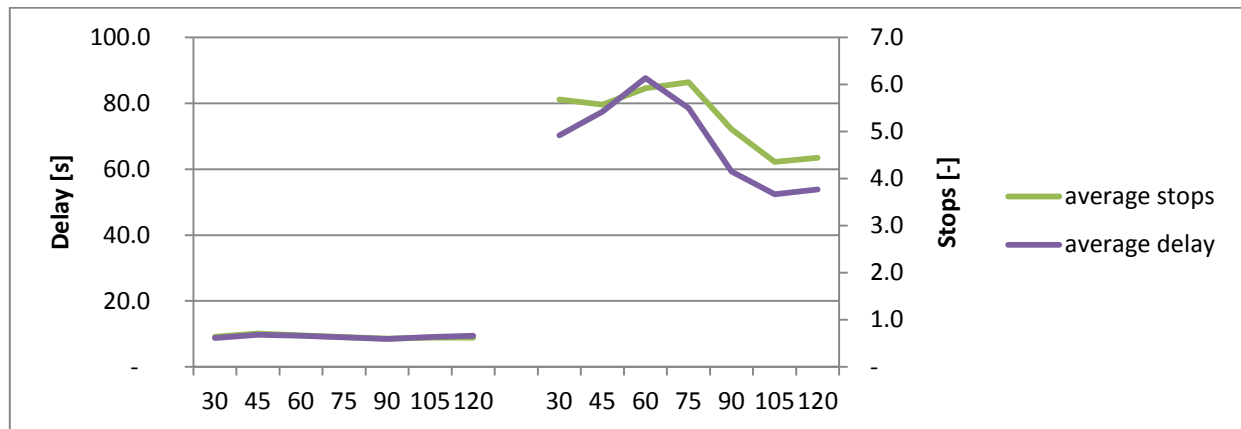


Figure 34: PIs for SWARM TLC 50% at Monza I4

Table 13: Relative PI changes of SWARM 50% against adaptive TLC at Monza I4

Time interval [min]	Off-peak		Peak	
	Stops [-]	Delay [s/veh]	Stops [-]	Delay [s/veh]
0-120	26%	14%	2%	1%
15-30	33%	25%	10%	6%
31-45	30%	19%	-4%	-8%
46-60	37%	27%	4%	6%
61-75	23%	10%	5%	-4%
76-90	30%	15%	-2%	-1%
91-105	6%	-4%	-4%	-4%
106-120	6%	3%	-1%	-5%

At the complex junction I3 again the base case (Figure 35) and adaptive ImFlow (Figure 36) have very similar values for both PIs. While SWARM improves immediately from 1% penetration rate onwards (Figure 37) both PIs during off-peak, this holds only for “average delay” during peak time. The number of stops exceeds at all penetration rates (Figure 38 for SWARM 50%) the ones of fixed time/adaptive.

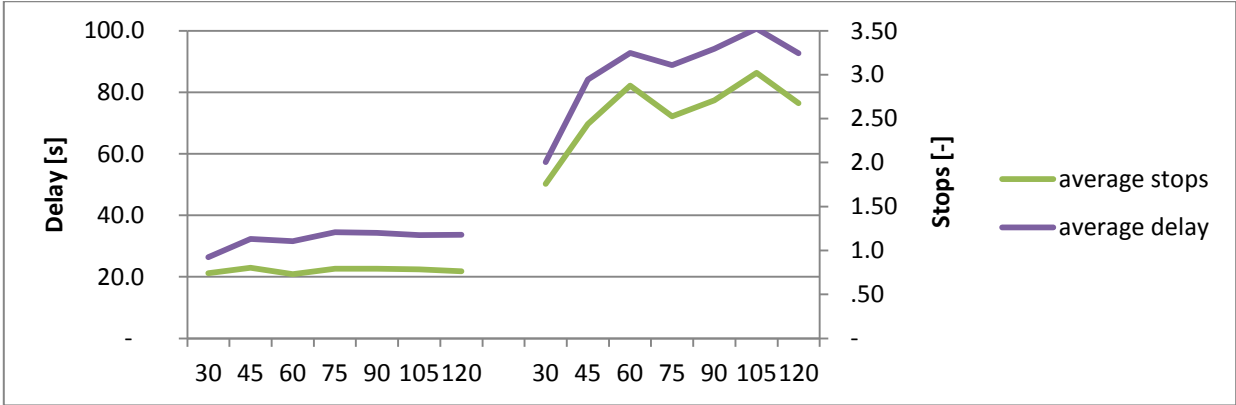


Figure 35: PIs for base case (fixed time TLC) at Monza I3

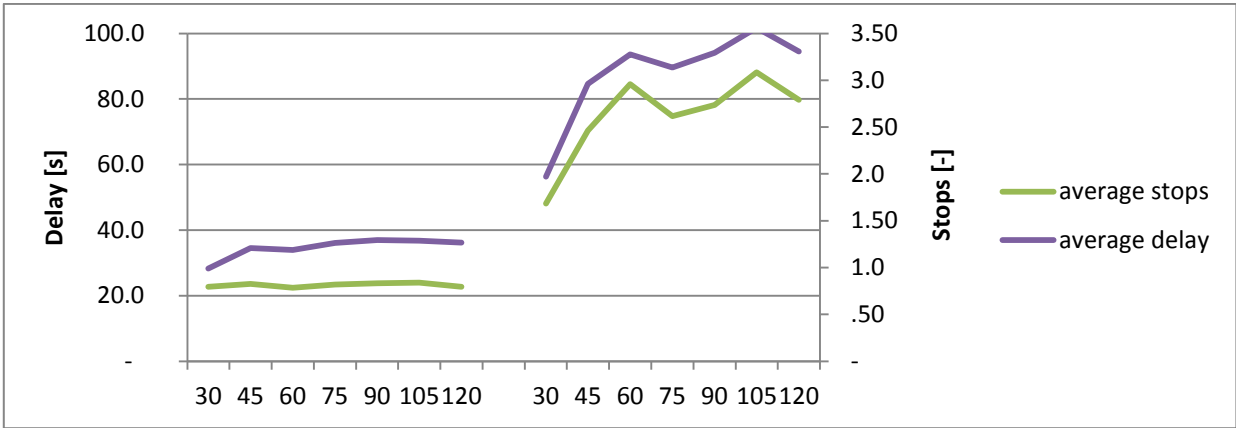


Figure 36: PIs for adaptive TLC at Monza I3

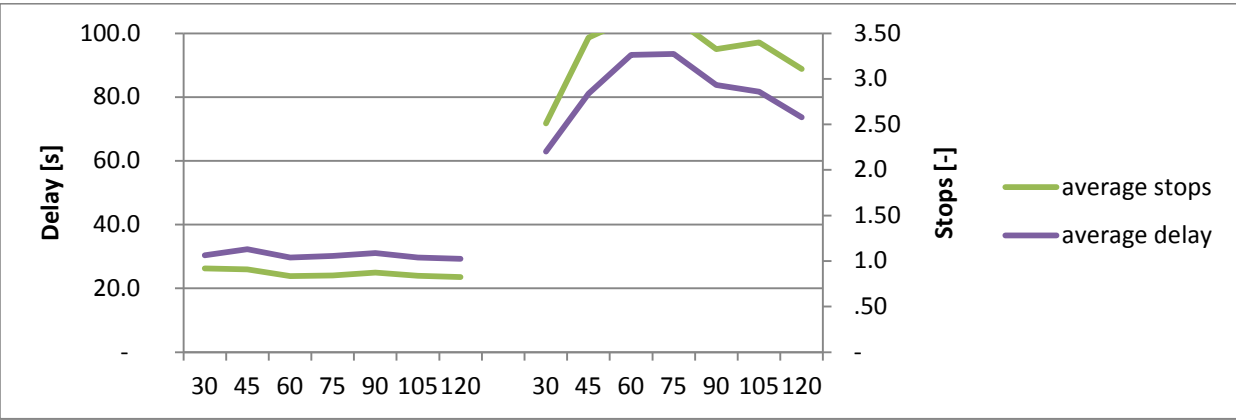


Figure 37: PIs for SWARM TLC 1% at Monza I3

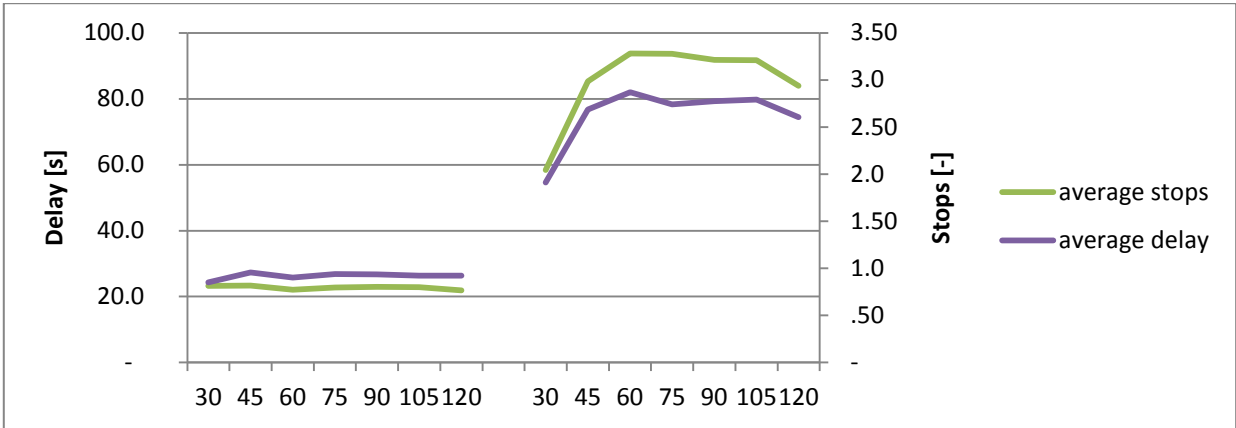


Figure 38: PIs for SWARM TLC 50% at Monza I3

**Table 14: Relative PI changes of SWARM 50% against adaptive TLC at Monza I3**

Time interval [min]	Off-peak		Peak	
	Stops [-]	Delay [s/veh]	Stops [-]	Delay [s/veh]
0-120	-1%	-23%	17%	-13%
15-30	2%	-14%	21%	-3%
31-45	-1%	-21%	21%	-9%
46-60	-2%	-24%	11%	-12%
61-75	-3%	-26%	25%	-13%
76-90	-3%	-28%	17%	-16%
91-105	-5%	-28%	4%	-22%
106-120	-4%	-27%	5%	-21%

### 5.3 Conclusion and Outlook

At the rather simple intersection I1 and complex junction I3 the SWARM algorithm can achieve improvements - even at very low penetration rates - compared to an already well-tuned adaptive TLC, namely ImFlow. At peak periods, however, this counts only for the PI “delay/waiting time”, while the number of stops might increase more or less. This can be explained by the different optimization procedures. While SWARM was tuned to minimize waiting time only, ImFlow takes both waiting time and stops in its objective function. This is why the stops are harder to defeat with cooperative control.

Intersection I4 is very simple and has a low traffic flow. Therefore ImFlow is able to run much longer cycle times as it can detect a maximum waiting time instead of keeping to a maximum cycle time (which may result in the traffic light giving green while there isn’t even someone waiting).

Traditional controllers have a large advantage at simple networks like RiLSA, as modelling the queues or flow is easy. On the large Monza intersection I3, however, there are multiple turn directions with separate lanes. When a vehicle passes the entry detection a traditional controller only assumes the probabilities for turning directions, based on historical averages. In reality there is statistical noise on the turning ratio and the estimates are off. This results in suboptimal control decisions, sometimes one queue even blocks the traffic flow of another causing a gridlock on the approach. Actuated has the same problem with turning vehicles, it may appear there is a gap in a flow, but the gap was caused by two vehicles in a row choosing a different turn direction while there is still a large platoon behind it. SWARM, on the other hand, looks at the entire approach and decides based on the flow there whether to cut off green or not.

The cost comparison with initial costs of 5,500€ and running costs of €350 p.a. towards the conventional loop detector solution at 12,000€ and 1,700€ p.a., respectively, promises good chances for this solution to be applied in reality.

Works are underway to combine the well-performing ImFlow adaptive control with the advantages of cooperative Traffic State Estimation (C-TSE) as already applied to SWARM. In first tests this setup outperforms the ones considered in this study.

## 6 Automatic configuration software

The traffic light control software developed within the COLOMBO project results in a large number of parameters that have to be set appropriately to optimize its performance. For the aim of finding performance-optimizing parameter settings, the COLOMBO project relies on the use of recent advances in the automatic configuration of algorithms and, in particular, on the usage of automatic algorithm configuration software. In fact, many modern optimization methods require the appropriate setting of a large number of parameters to optimize their performance and automatic algorithm configuration tools have a pivotal role to tune even rather complex algorithms with tens or hundreds of algorithm parameters. Often, the so obtained parameter settings surpass the results achieved by human designers. We have explored within the COLOMBO project several automatic configuration software tools for the novel usage of automatic algorithm configurators to tune traffic light control software. Essentially, when an automatic algorithm configurator is used for this task, a simulation-optimization loop results, where simulation of the traffic is used to provide an estimate of the performance of the parameter settings of the traffic light control software. As the specific automatic algorithm configurator we have used and also further improved the irace software [López-Ibáñez et al., 2011], which was found to be the most useful automatic algorithm configurator for the typical tasks arising in the COLOMBO project. As shown by the results presented in the COLOMBO project (in particular, in deliverables D3.2 and D3.3), this overall simulation-optimization approach can be fruitfully applied to tune traffic light control software such as the SWARM algorithm and below a short summary of the main findings is given. In the following, explain basic details about how to set up the simulation-optimization loop, give a summary of the results of deliverables D3.2 and D3.3 and discuss some recent results obtained with a slightly different setup.

### 6.1 Simulation setup

#### 6.1.1 Tuning overview

The SWARM traffic light control system that has been developed in the COLOMBO project has multiple configuration parameters, allowing a better adaptation to different road networks and traffic flows configurations. Since the number of such parameters is usually over 100, an automatic tuning software is better suited to search for a high-quality configuration. The system used for the automatic tuning is irace [López-Ibáñez et al., 2011].

To obtain a configuration for the swarm traffic light controller with the automatic tuning software a tuning setup has to be devised. The setup specifies:

- on which road network the swarm controller has to work on;
- the set of traffic flows that have to be simulated in the road network to evaluate the performance of a configuration;
- the budget allocated for the tuning (this can be thought as the total number of simulations that will be executed during the tuning process).

By using an appropriate set of traffic flows, both the time span of the simulation and the equipment penetration rate can be set.

The traffic flows for a particular road network are generated using a script that relies on realistic load curves and equipment rate. In the tuning phase, for each of the considered penetration rates 100%, 50%, 25%, 10%, 5%, 2.5%, and 1% a set of 1,000 different flow instances was used.

Both the tuning software and the evaluation phase require a way to evaluate a particular configuration. The metric adopted here is the average waiting time of the cars. The waiting time of a car represents the total number of seconds the vehicle is considered stopped by the simulator. To obtain the average value we compute the sum of all the waiting times of the vehicles and we divide it by the number of cars simulated.



At the end of a tuning execution, when the allocated budget has been spent, the tuning software returns the set of best configurations found. The set is usually composed of eight different configurations. As mentioned above, the tuning algorithm uses a specified road network. Different road networks usually require different tuning configurations. The evaluation phase of the best configuration obtained can use the same scenario or a different one.

To explore these circumstances we used a configuration obtained with a particular tuning setup on different scenarios, changing the road network and the traffic flows used.

### 6.1.2 Evaluation overview

To evaluate a configuration, it is necessary to specify the road network and set of traffic flows. As before, the traffic flows specify the simulation time span and the equipment penetration rate. In the evaluation phase we generate new traffic flows instead of using the ones that are used during the tuning. This separation between flows used in the tuning and the evaluation follows the separation in training and test data usually done in machine learning.

Since each flow instance is different, some could be more favourable than others, and this could lead to flow sets that are on average more or less “difficult” for the traffic light. This behavior can be best observed in the evaluation of the static and the actuated traffic light controller, and it is particularly relevant when assessing the performance of the swarm controller for different penetration rates. To avoid uncertainty over whatever the result difference is caused by the penetration rate change or the different set of traffic flows, we decided to use the same instances for every rate, changing only the vehicle types to properly reflect the penetration rate wanted. This guarantees that different results are only due to the penetration rate change and not caused by the traffic flow itself.

The default behaviour of SUMO is to teleport vehicles that are stopped for more than a time threshold. While this is generally a reasonable behaviour, when evaluating a traffic light controller, it can lead to skewed results since some vehicles can move in an unrealistic way. Hence, we have then decided to disable this behaviour. Unfortunately, such decision could (rarely) cause deadlocks during the simulation (in case of a feasible but of poor quality configuration). Consequently, we monitor the time used by SUMO for a simulation and stop its execution if it goes over a threshold, configured to be significantly larger than the normal execution time.

For details of how the irace software needs to be interfaced with the simulation of the traffic, we refer to [COLOMBO D3.2, 2014].

Next we first summarize the main findings that were obtained in the evaluation of the offline and online tuning strategies developed within WP 3, focusing on the most recent results of deliverable D3.3.

## 6.2 Findings on automatic configuration for the initial design

The most important conclusion from the tuning in deliverable D3.2 and in part of D3.3 is that the automatic configuration of the SWARM traffic light control software by irace is effective and that high-performance configurations can be found. For obtaining well performing default configuration of the SWARM software, it seems sufficient to use a tuning on a single intersection that is controlled by it. The irace software was able to find parameter settings that improved strongly over initial hand-tuned settings or random configurations that were used as an additional baseline comparison. The automatic configuration process was also reasonably efficient. For example, the execution of 50,000 fine-grained simulations of a 24 hours traffic and a single intersection controlled by the SWARM traffic light took approximately 60 CPU days on a single core of an AMD Opteron 6272 CPUs running at 2.1 GHz; using a straightforward parallelization by running simulations in parallel, a linear speed-up can be achieved and using a moderate size cluster computing infrastructure simulation times well below one day have usually been achieved.

## 6.3 Findings on automatic configuration for online adaptation

For obtaining an initial default configuration by automatic configuration software, specific assumptions on the environment have to be made such as assuming a specific traffic density or road topology. Such a default configuration also works reasonably well when transferred to other environments. However, maybe as expected, further improvements are possible by taking into account specific characteristics of the environment in which the controller will be used and by re-optimizing the controller behavior. The results from deliverable D3.3 have shown that such an online adaptation to specific environments is beneficial once the traffic light controller is put into a production environment both for (i) reaching better performance, and (ii) adapting itself to changes of the characteristics of the traffic (density, penetration rate) or of the road network topology around the intersection. In particular, the following main trends have been observed during our experimental campaign and should, hence, be taken into account by practitioners.

### 6.3.1 Change of the penetration rate

A decreased penetration rate seems to significantly impact the performance of the SWARM traffic light controller that has been configured assuming a higher penetration rate. This observation has some importance in the case of equipment failure leading to the non-detection of equipped cars, which effectively corresponds to a lower penetration rate down to 0% in case of complete failure. Otherwise, we assume a decrease in the penetration rate is rather unlikely to happen in practice. On the other hand, an increased penetration rate does not impact too strongly the performance of the SWARM traffic light controller that was configured assuming a lower one. This suggests that the SWARM algorithm can operate for a certain time, even with assumed penetration rates that are lower than the actual ones, and provide reasonably high performance before a re-optimization is advisable.

### 6.3.2 Change of TLC parameters

When moving from one traffic light scenario such as a single SWARM controlled intersection to a different one, for example a corridor controlled by SWARM traffic lights, a re-optimization of the SWARM parameters was found to be useful. Specific results have been presented in deliverable D3.3 for the change to a corridor of three traffic lights, which are arranged in sequence, controlled by the SWARM software. Depending on the penetration rate, improvements between 2% to 15% have been observed by a rather limited re-optimization of the SWARM controller. The general tendency here is that the improvement is largest for a small penetration rate and that it reduces for higher penetration rates. For larger penetration rate, the performance is more robust to some changes of the penetration rate.

### 6.3.3 Change of the traffic density

We have also considered a change in the traffic density.<sup>4</sup> The experimental results we have obtained suggest that the performance of SWARM is rather robust with respect to smaller changes in the traffic density of up to about 20%. Our conclusion here is that a change of the overall traffic density requires a re-optimization only once these changes are becoming rather large.

## 6.4 Extended experiments

In follow-up experiments, we have considered an extended setup of the experiments, where we used the one hour traffic flows for the tuning and then evaluate the resulting configurations in different environments. For the tuning we used the one hour traffic flows as the resulting tuning times are by at least a factor 10 smaller than when basing tuning on the 24 hour simulation. Still, the tuning is a

<sup>4</sup> Note that this refers to a global increase of traffic throughout a whole day. Small peaks due to accidents or other specific incidents are handled directly by the SWARM algorithm's macro policy through switches between different micro policies.

time consuming procedure, and it is therefore worthwhile to evaluate the controller on a scenario even before tuning on it, using, for instance, a configuration computed for a different road network. The extended experiments show that in some cases the performance of the controller on different scenarios (but tuned with the same parameters) are comparable.

Table 15 reports the experimental set up adopted for the evaluation of the swarm controller. Each column represents a different scenario, while the rows are the applied tuning setup. Note that each row of the table represents a set of seven different tunings, one for each penetration rate considered. As mentioned in the previous section, after a tuning execution multiple best configurations are obtained as a result. In the evaluation phase, we evaluate each of them since the first one is not necessarily the best one. As this leads to a large amount of simulations and results, in the comparison charts shown below and in the appendix 1, for every tuning setup we plot only the best results obtained from the configuration that performs best.

The traffic flows set also the time duration of a simulation, which are 1 or 24 hours for the RiLSA, Corridor and Grid network and 2 hours for Monza. The total simulation time is usually longer than what a particular flow specifies, since the last cars that entered the network will need some time to conclude their trip. Sumo can limit the simulation time to an exact instant, but this could lead to an unfair evaluation of the traffic light controllers since a different number of cars could have passed the intersection at the selected end instant. To avoid this possibility we decided to terminate the simulation when all the cars complete their paths and exit the simulation.

In each case, apart from Monza, the traffic flows used to obtain the best configurations through the automatic tuning were for one hour of traffic. In the Monza scenario the flows were two hours long.

**Table 15: Evaluation plan**

Tested on Tuned on & budget		Rilsa 1h	Rilsa 24h	Grid 1h	Grid 24h	Corridor 1h	Corridor 24h	Monza 100% density	Monza 60% density
Rilsa	20k	X	X	X	X	X	X	X	X
Rilsa	100k	X	X	X	X	X	X	X	X
Rilsa	250k	X	X	X	X	X	X	X	X
Rilsa	500k	X	X	X	X	X	X	X	X
Grid	20k	X	X	X	X	X	X		
Grid	50k	X	X	X	X	X	X		
Grid	100k	X	X	X	X	X	X	X	X
Corridor	20k	X	X	X	X	X	X	X	X
Corridor	50k	X	X	X	X	X	X	X	X
Corridor	100k	X	X	X	X	X	X	X	X
Monza	20k							X	X

## 6.5 Results

For space reasons, all the results are reported the appendix A, Figure A to Figure H. In this section we present a subset of the results.

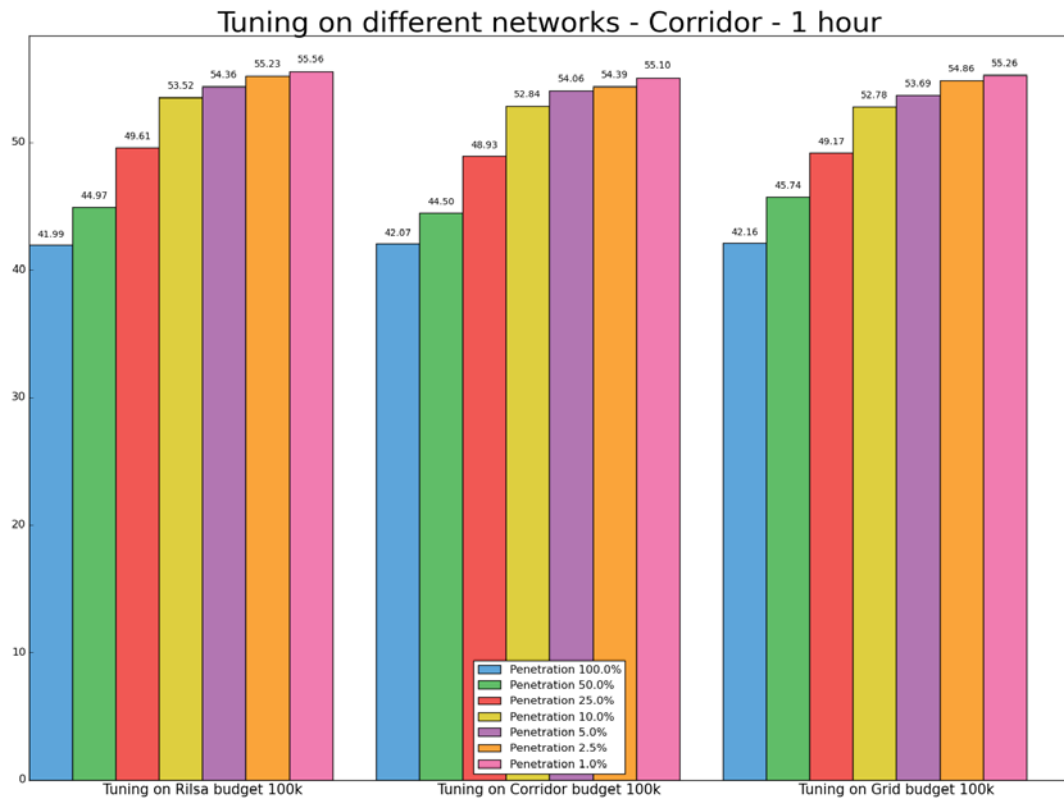
In the charts, the x-axis indicates the tuning setup while the y-axis is the average waiting time of the vehicles. Each bar of a setup represents the best result obtained among the set of best configurations found by the tuning process. Each of these plotted values is the average of the results of 100 simulations with different traffic flows. The colour of the bars represents the penetration rate used.

As stated before, the different penetration rates have the same set of flows: in every chart the values of the actuated and static traffic light controllers are equal.

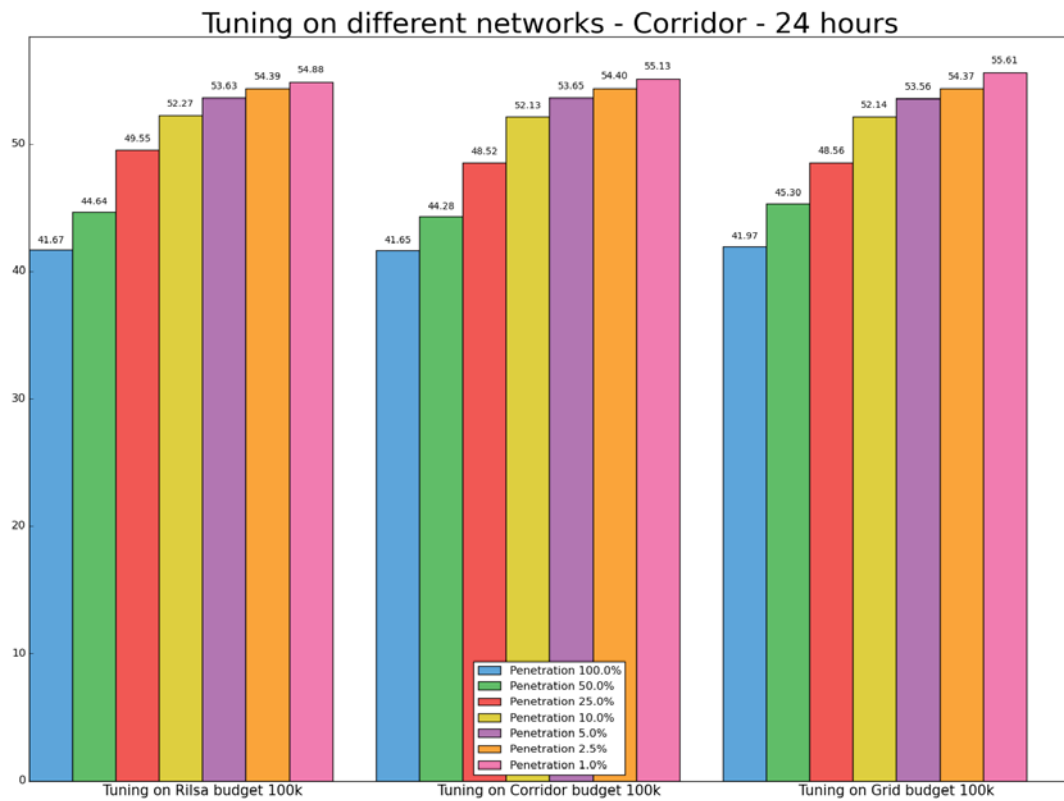
### **6.5.1 Same tuning on different networks**

A configuration set obtained on a particular road network can also be evaluated on different ones. From the evaluation that we have conducted we can observe that the swarm controller is usually quite robust to the changes of road network, since the results are comparable regardless the road network used by the tuning process. (Note that these results here may differ from the ones in deliverable D3.3, as in D3.3 a single configuration was evaluated and our findings here refer to the best configuration among a candidate set of elite configurations that is identified by a re-evaluation on a different scenario.

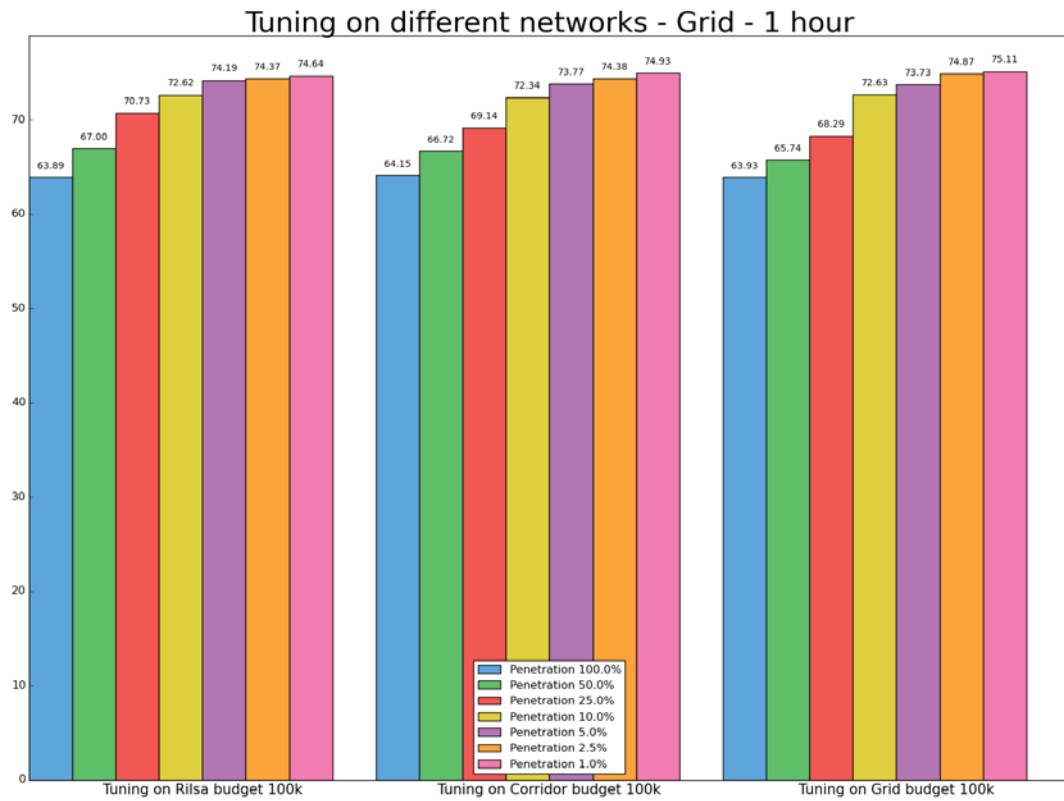
The figures below show this behaviour quite well. In the Monza scenario (Figure 42), the results are not as aligned as in the other cases, but this is due to the fact that Monza has a road configuration that is different w.r.t the RiLSA, the Corridor, and the Grid.



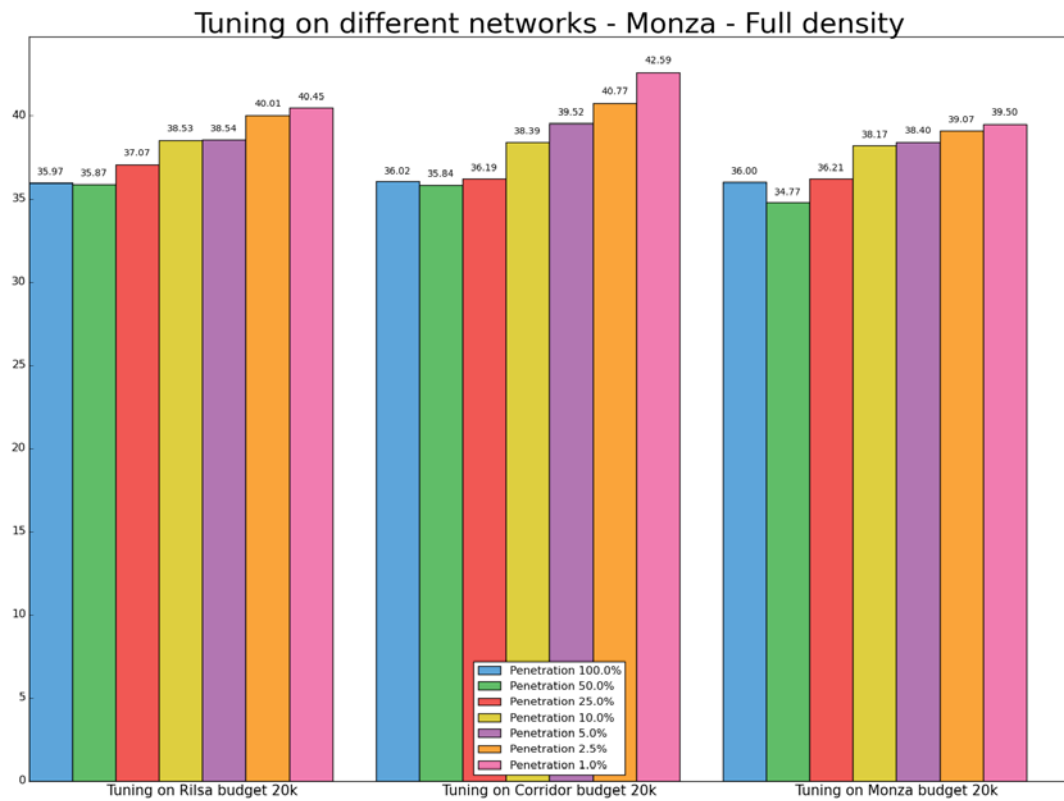
**Figure 39: Tuning on RiLSA, Corridor and Grid with the same budget evaluated on Corridor 1h.**



**Figure 40: Tuning on RiLSA, Corridor and Grid with the same budget evaluated on Corridor 24h.**



**Figure 41: Tuning on RiLSA, Corridor and Grid with the same budget evaluated on Grid 1h.**

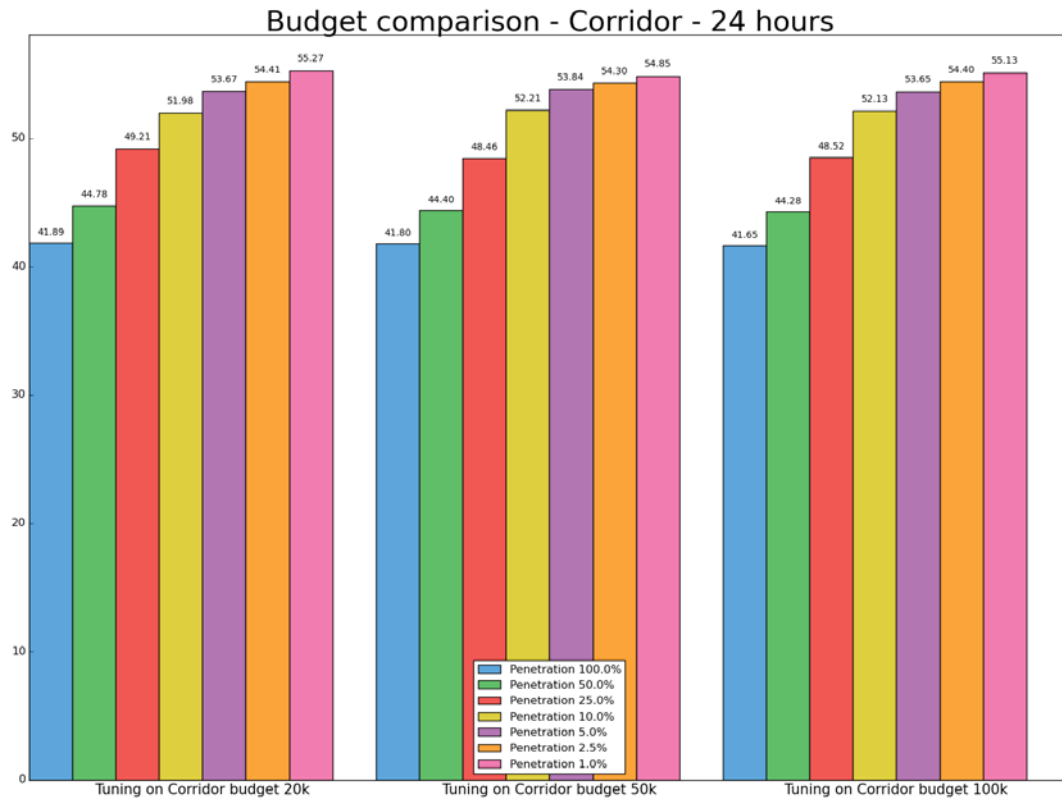


**Figure 42: Tuning on RiLSA, Corridor and Monza with the same budget evaluated on Monza**

## 6.5.2 Tuning budget influence

In the different evaluations that we have conducted we have not found a clear correlation between the tuning budget used in the tuning process and the performance of the system.

In the following charts, we highlight the different results obtained with increasing budget. As it can be seen, the outcomes are comparable in all cases, without significant deviation between the different results, even after large change in allocated budget.



**Figure 43: Budget comparison on Corridor 24h for 20k, 50k and 100k.**

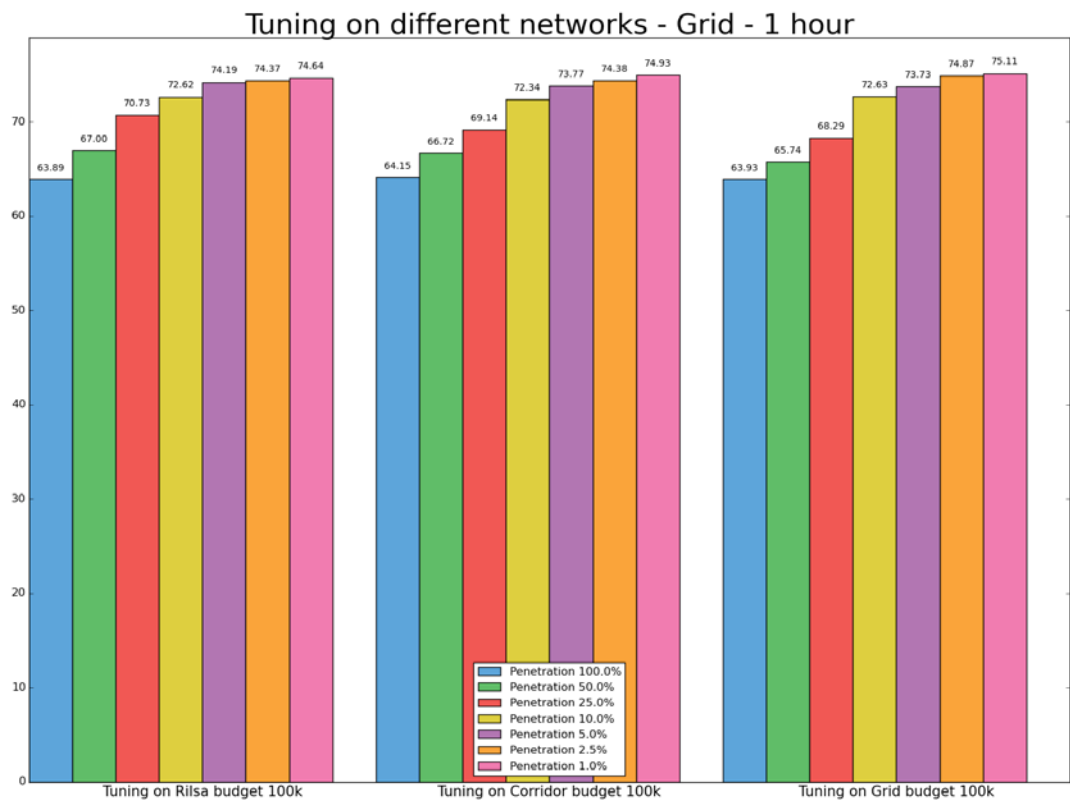


Figure 44: Budget comparison on Grid 1h for 20k, 50k and 100k.

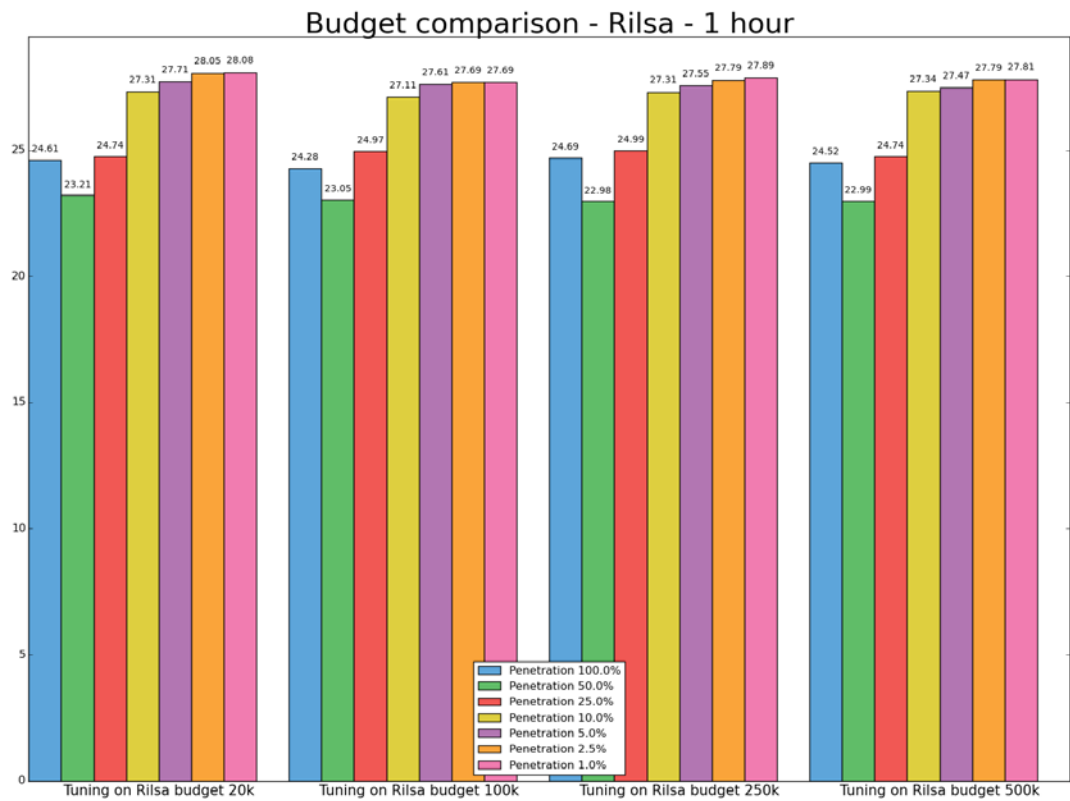


Figure 45: Budget comparison on RiLSA 1h for 20k, 100k, 250k and 500k.



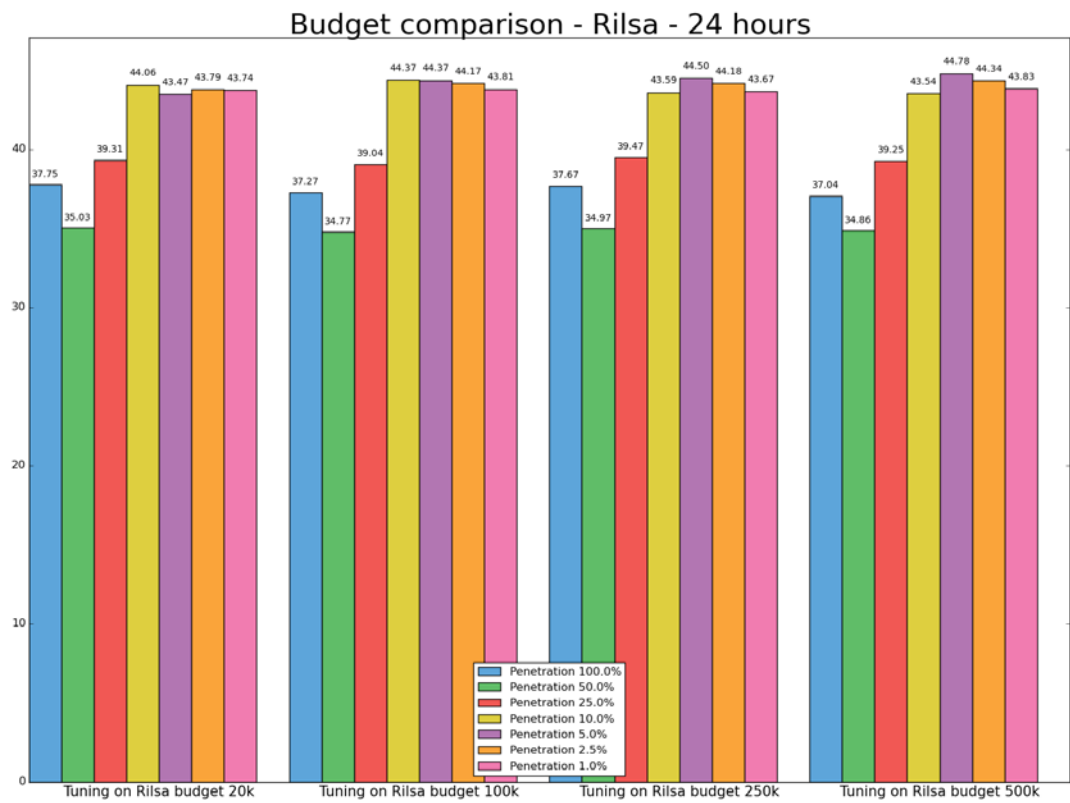


Figure 46: Budget comparison on RiLSA 24h for 20k, 100k, 250k and 500k.

## 7 Communication Integration

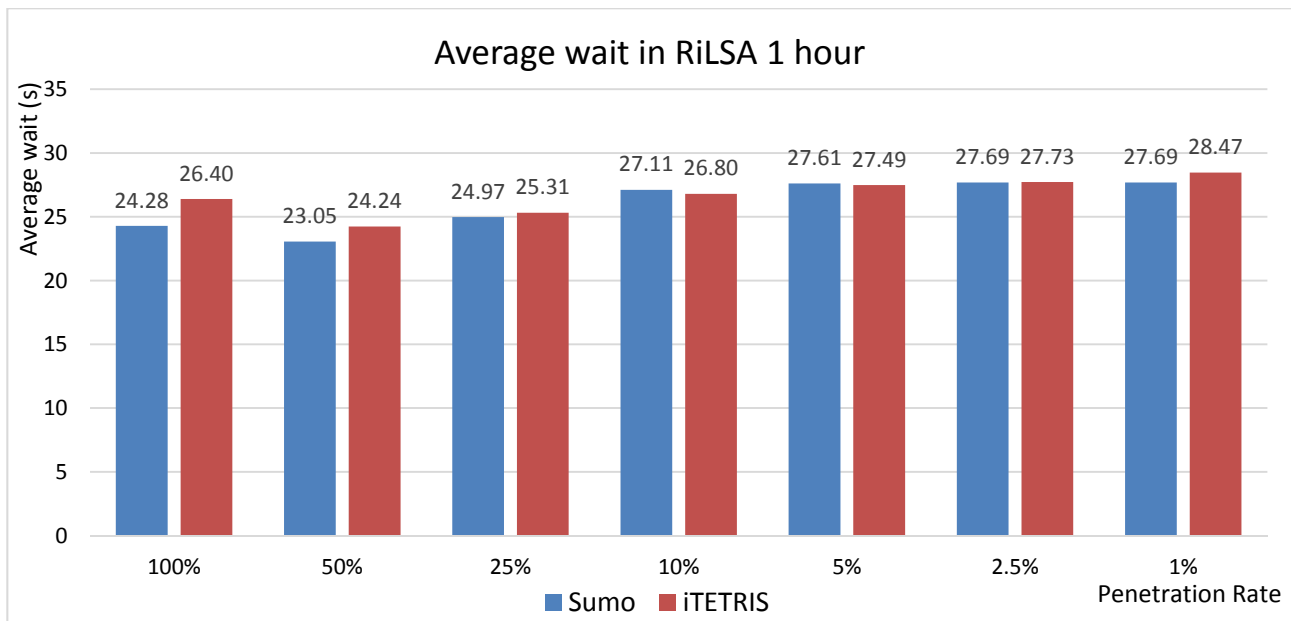
This final set of results compares the PI “average waiting time” obtained with two different simulator configurations: with and without integration of iCS (iTetris).

The standalone SUMO implementation uses every car that it can see (i.e. the car is simulated with the transmission equipment) to gather the information. The underlying hypothesis of having full knowledge on traffic is not realistic, since a monitoring protocol cannot have a perfect knowledge on the traffic state.

The integration in iCS better represents a real environment, since the information used by the traffic light controller is not collected directly from SUMO, but it comes from the monitoring protocol that uses ns-3 to simulate the message communications. For the simulations in iCS we used the UNIBO monitoring protocol and the swarm traffic light controller, and repeated each experiment over 33 different route flows.

The configuration parameter for the swarm TLC and the traffic flows are the same in the two simulation systems.

Figure 47 and Table 16 show the results of both system configurations. For higher penetration rates, the results are better in the standalone SUMO system, while for lower penetration rates they are comparable. This discrepancy can be explained by the lower quality of the information available to the controller in the iCS due to the realistic monitoring protocol used to collect them. This difference is more pronounced when a lot of information is available (i.e. in high penetration rates), while it has a lower influence when only a fraction of vehicles are visible in the system, since the controller is tuned to be less reliant on them.



**Figure 47: Comparison of the average wait in RiLSA for the standalone SUMO system and the iTetris integration.**

**Table 16: Comparison of the systems**

Penetration rate	SUMO	iTetris	Difference iTetris vs SUMO
100%	24.28	26.40	8.73%
50%	23.05	24.24	5.16%
25%	24.97	25.31	1.36%
10%	27.11	26.80	-1.14%
5%	27.61	27.49	-0.43%
2.5%	27.69	27.73	0.14%
1%	27.69	28.47	2.82%

The encouraging results obtained show the importance of automatic configuration methods in the development of optimization and control software and we believe that the resulting simulation-optimization cycle will be in the future more frequently being used in the industry. We therefore recommend practitioners to educate themselves on these techniques, and to consider making use of them.

## 8 References

- [CST, 2012] Centro Studi Traffico / Provincia di Monza e Brianza: Studio per la Definizione e Valutazione del Nuovo Assetto della SP 13 tra Colleoni e Pompei, Milano (IT), 2012.
- [eCoMove, 2013] eCoMove project, Simulation results for validation and impact assessment, Deliverable D5.10 (restricted), 2013.
- [ISO, 2015] International Organization for Standardization: ISO/TR 16786:2015-08 (TECHNICAL REPORT) Intelligent transport systems - The use of simulation models for evaluation of traffic management systems - Input parameters and reporting template for simulation of traffic signal control systems, Vernier (CH), 2015.
- [Katsaros et al., 2011] Konstantinos Katsaros, Ralf Kernchen, Mehrdad Dianati, David Rieck: Performance study of a Green Light Optimized Speed Advisory (GLOSA) Application Using an Integrated Cooperative ITS Simulation Platform, IEEE, 2011.
- [López-Ibáñez et al., 2011] Manuel López-Ibáñez, Jérémie Dubois-Lacoste, Thomas Stützle, and Mauro Birattari: The irace package, iterated race for automatic algorithm configuration. Technical Report TR/IRIDIA/2011-004, IRIDIA, Université Libre de Bruxelles (BE), 2011.
- [Van Vliet, Turksma, 2013] Koos van Vliet, Siebe Turksma: ImFlow: Policy-based adaptive urban traffic control - First field experience, TO0231, 9<sup>th</sup> ITS European Congress, Dublin (IE), 2013.

## Appendix A -Tuning Results

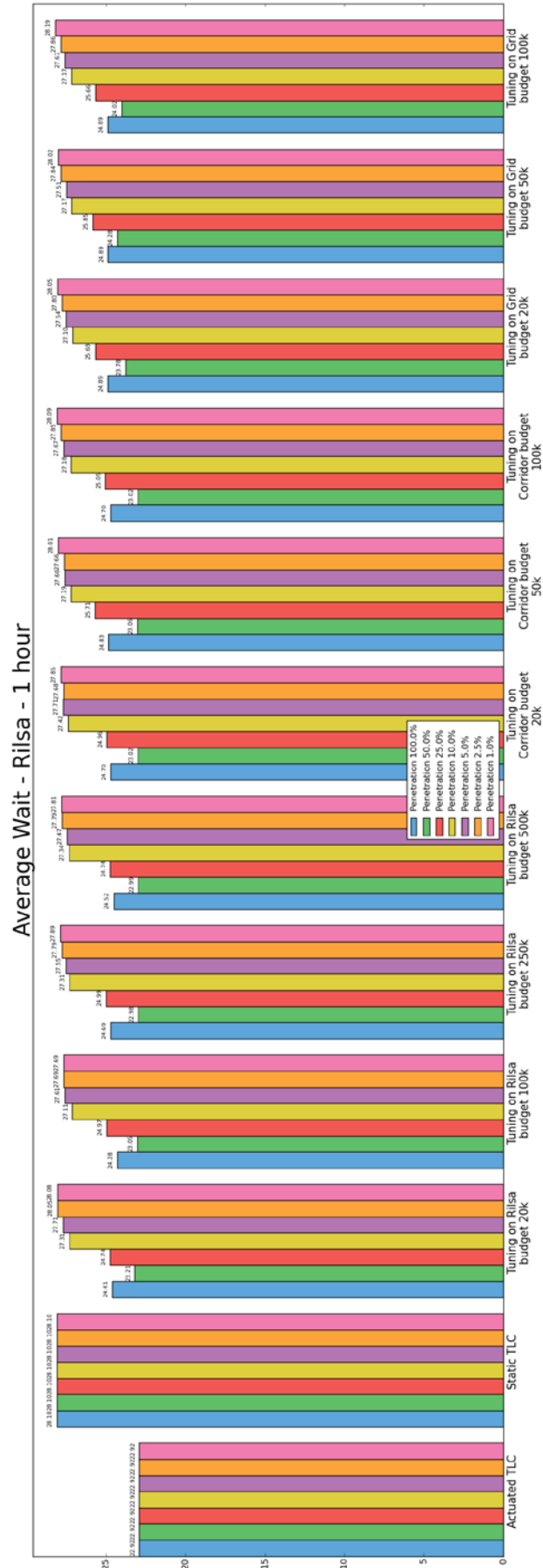


Figure A: Complete results for RiLSA 1 hour

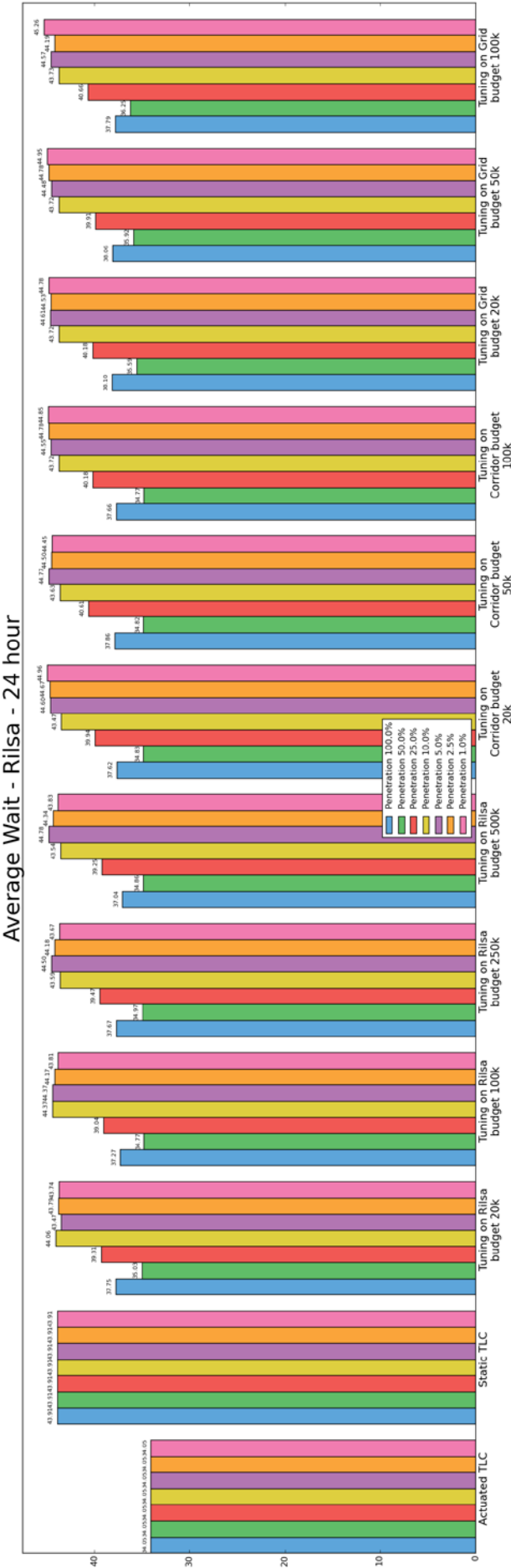


Figure B: Complete results for RiLSA 24 hours

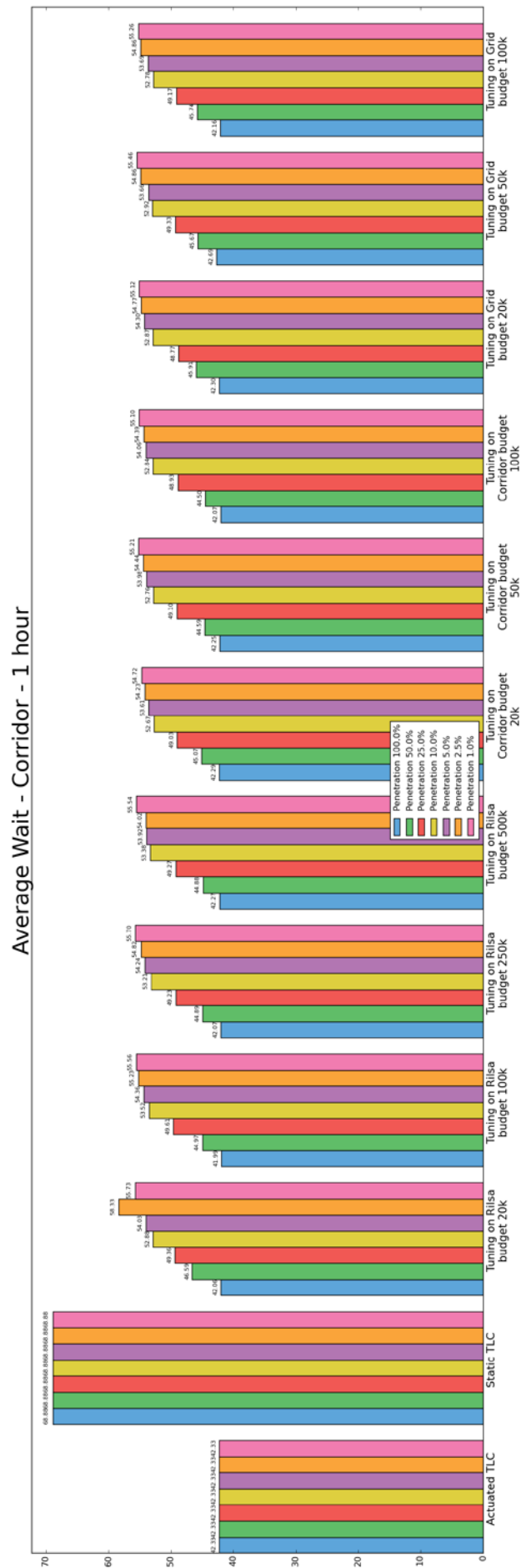


Figure C: Complete results for Corridor 1 hour



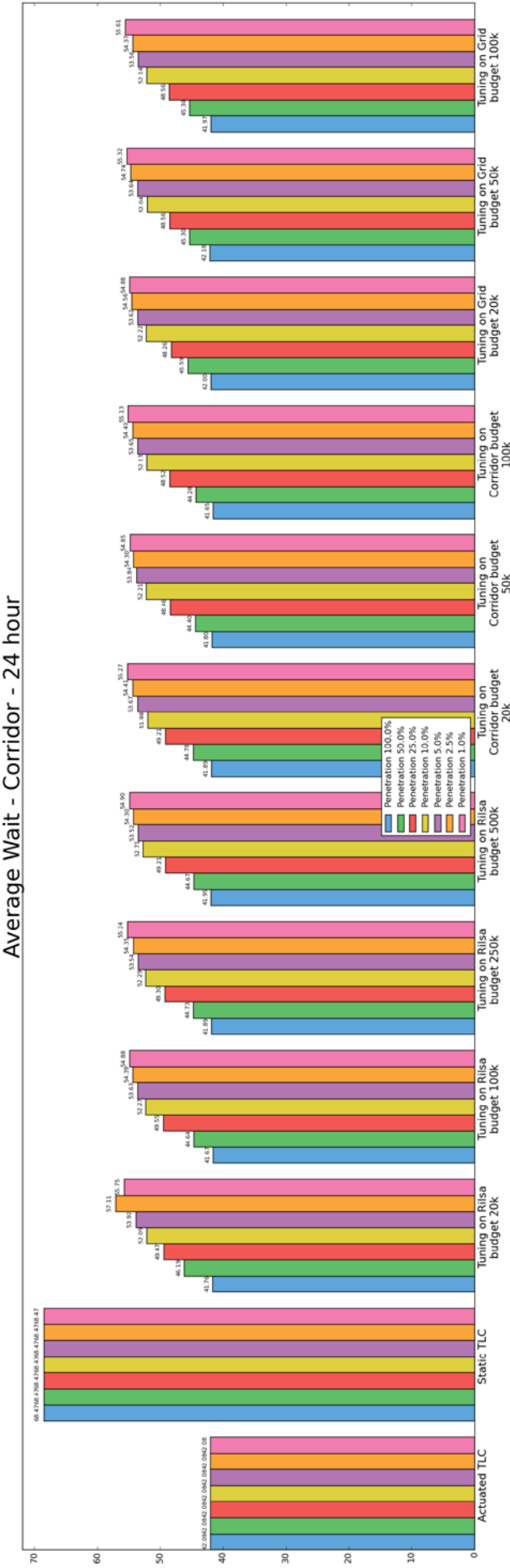


Figure D: Complete results for Corridor 24 hours

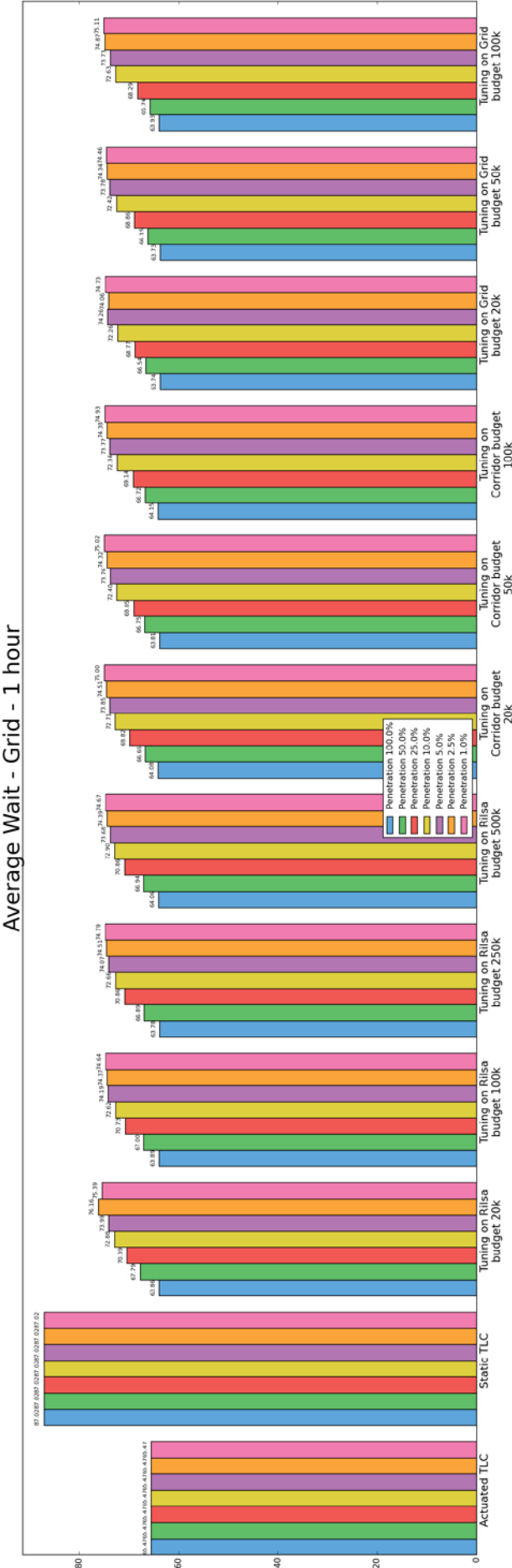


Figure E: Complete results for Grid 1 hour

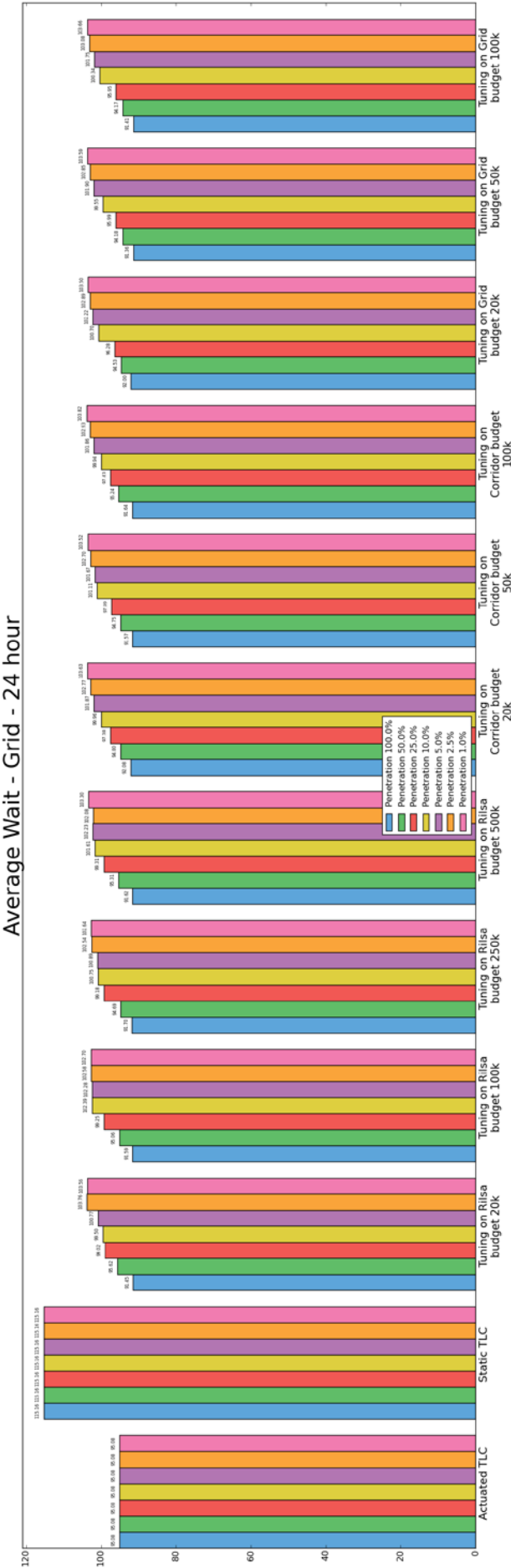


Figure F: Complete results for Grid 24 hours

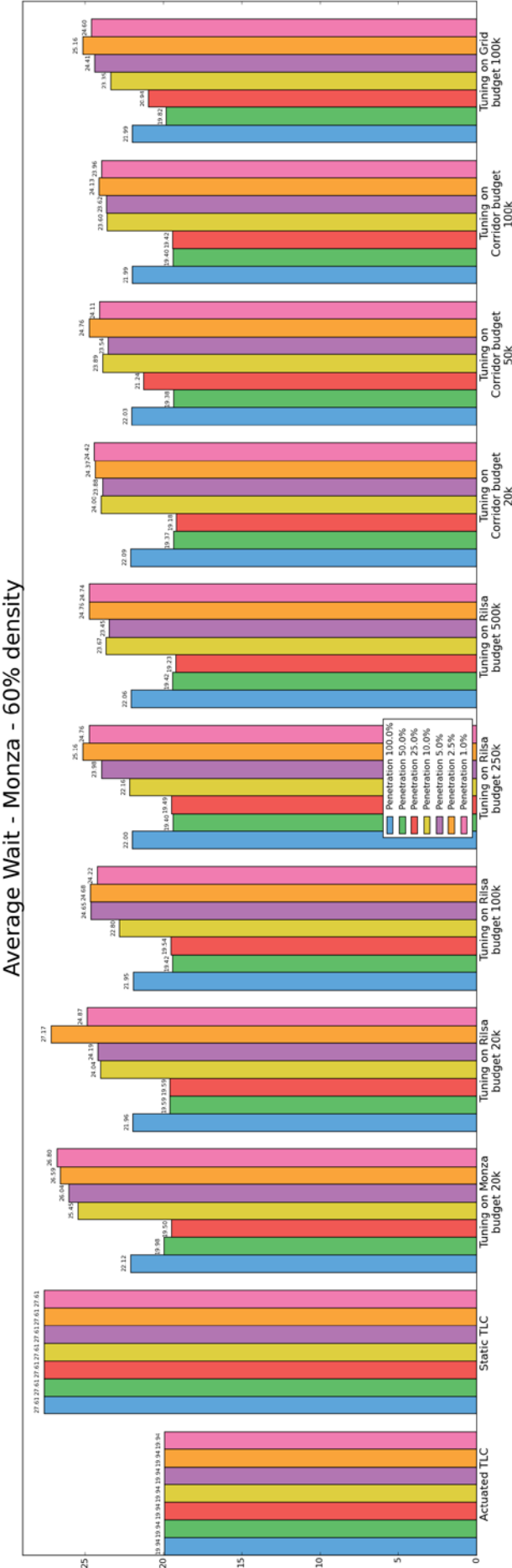


Figure G: Complete results for Monza 60% density

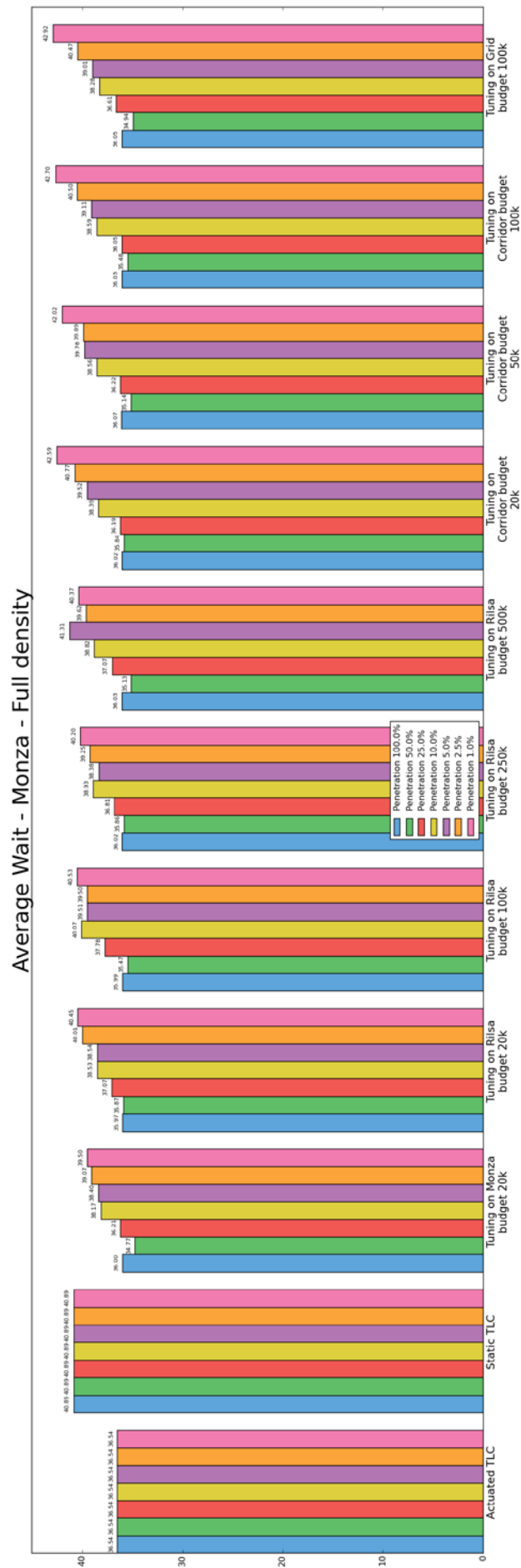
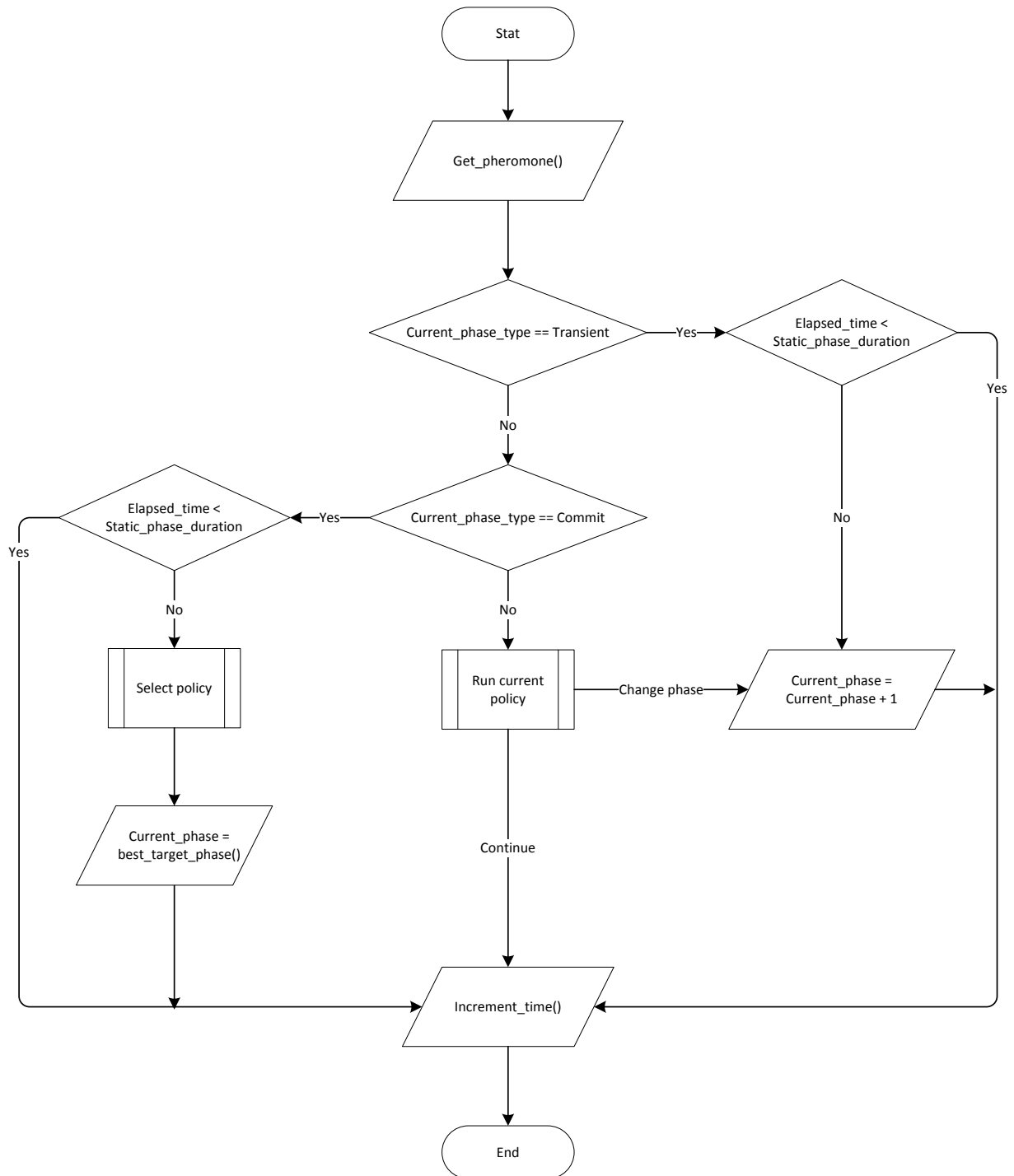


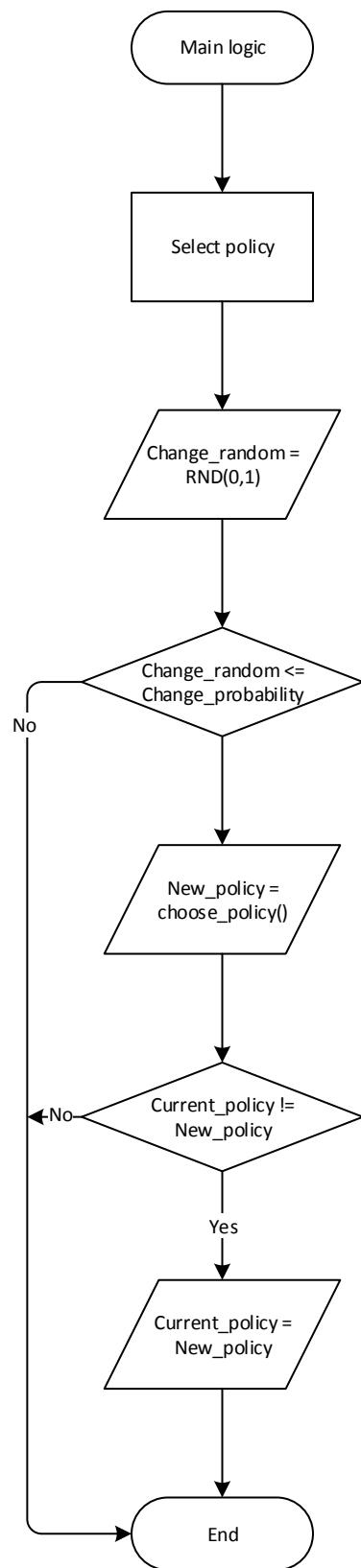
Figure H: Complete results for Monza 100% density

## Appendix B – SWARM algorithm flow charts

### Main logic

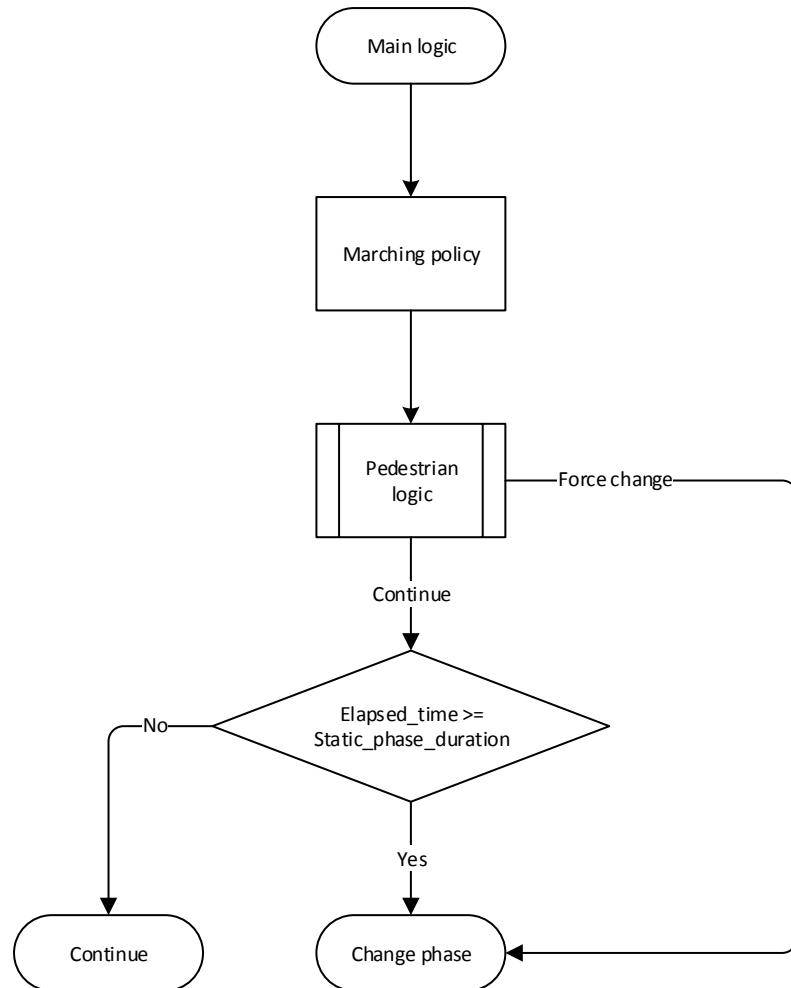


## Select policy

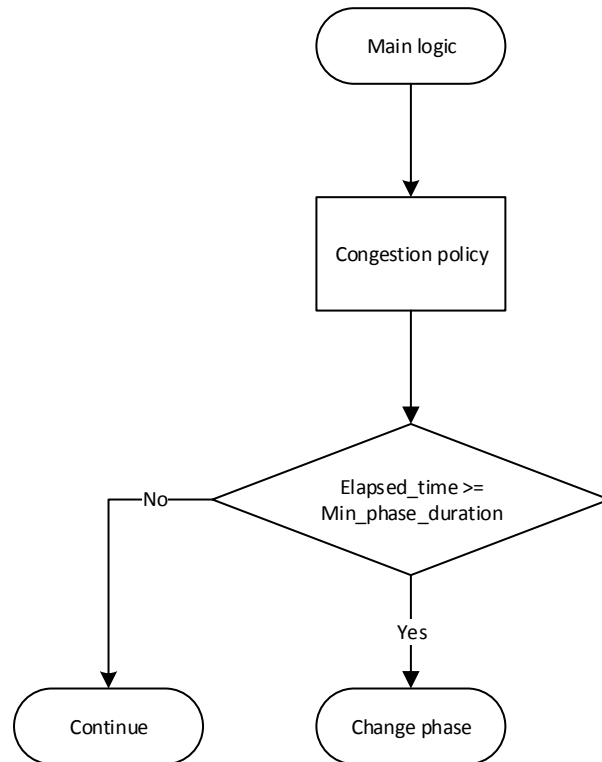




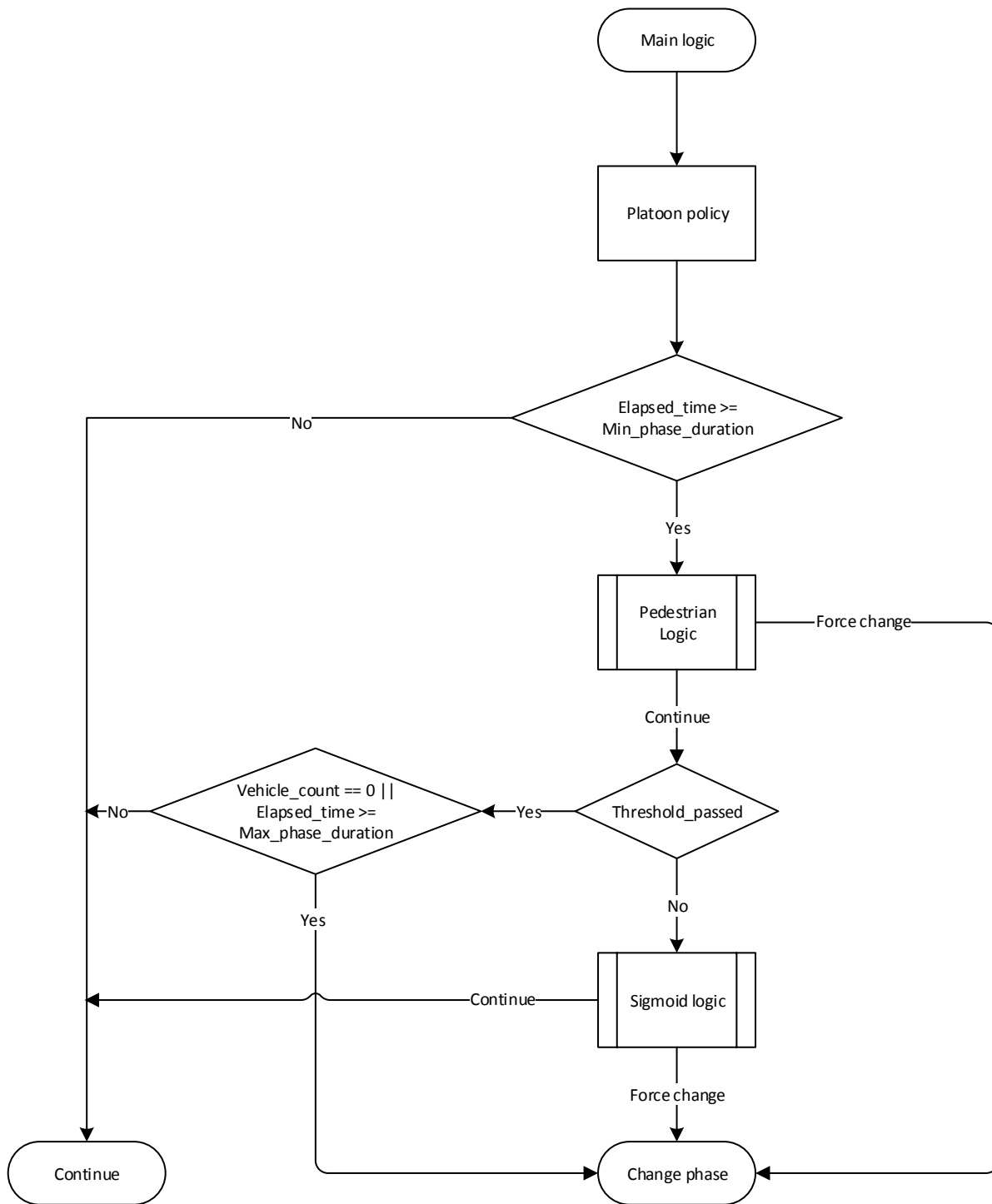
# Marching Policy



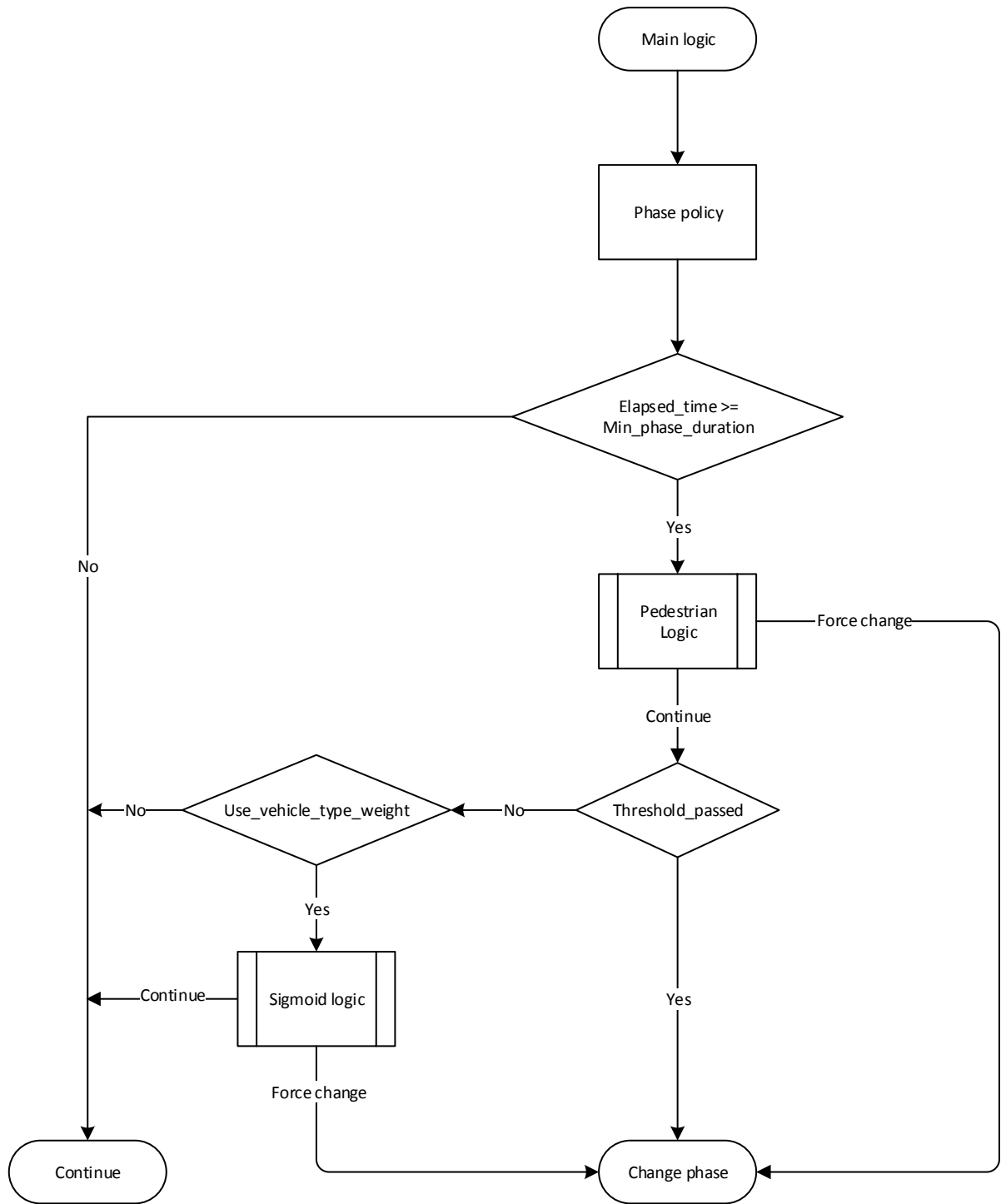
## Congestion Policy



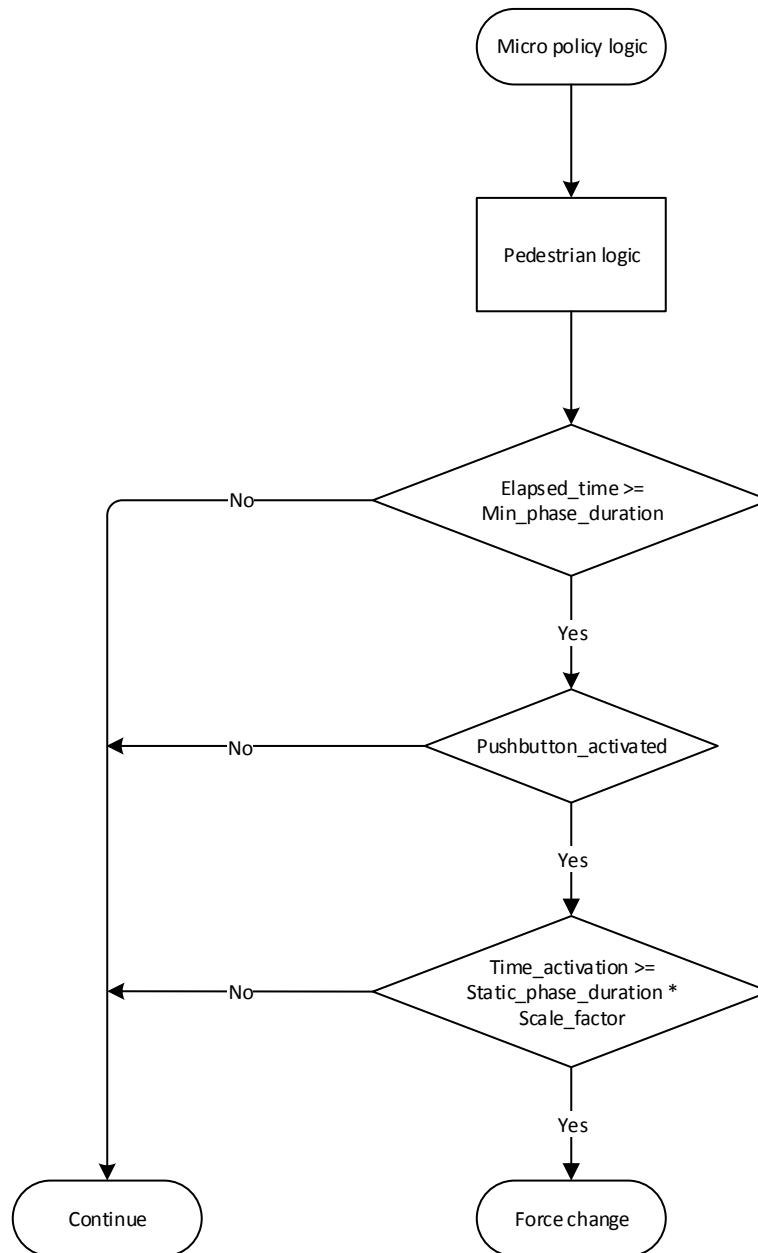
# Platoon Policy



# Phase Policy



## Pedestrian Logic



## Sigmoid Logic

